



Universidad de las Ciencias Informáticas

Facultad 4

“
Sistema de gestión de información para el Departamento de opinión pública del
”
PCC de Cienfuegos

Trabajo de diploma para optar por el título de Ingeniero en Ciencias
Informáticas

Autores:

Orlay Gainza Heredia

Osmar Núñez Rodríguez

Tutor (es):

Ms.C Ing. Rosalba Carralero Medina

Dr. Rafael Rodríguez Puente

Co-Tutor (es):

Ing. Gustavo Crespo Sánchez

La Habana, junio de 2015.

“Año 57 de la Revolución”.

Declaración de Autoría

Declaración de Autoría

Declaramos ser los autores del presente trabajo de diploma y otorgamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Firma del autor
Orlay Gainza Heredia

Firma del autor
Osmar Núñez Rodríguez

Firma del tutor
Ms.C. Ing. Rosalba Carralero Medina

Firma del tutor
Dr. Rafael Rodríguez Puente

Firma del co-tutor
Ing. Gustavo Crespo Sánchez



“... somos fuertes, realmente fuertes, porque contamos con la opinión pública del país, que es un arma más poderosa que ninguna otra...”

Fidel Castro Rus. 1986

De Orlay:

Dedico este trabajo a 3 personas que son muy importantes para mí, a mi abuela (Marta) y mi mamá (Margot) que son mi vida y se han sacrificado para que pudiera lograr mis sueños, a ti abuelita por apoyarme estando al lado de mami, a mi novio (Ovidio) por su amor incondicional sin pedir nada a cambio, a él que se preocupó por mis problemas y me motivó en cada momento cuando creía que todo estaba perdido, es por ellos que pude lograrlo.

De Osmar:

Con todo cariño y amor dedico este trabajo a 4 personas en especial, a mi mamá (Nidia) y a mi papá (Osmel) que han sacrificado todo por mí y han estado siempre ahí cuando los he necesitado, a mi tía (Gladis) y mi abuela (Ocilia) que siempre me apoyaron y confiaron en mí todo el tiempo, gracias a ellos pude lograr esta meta.

Agradecimientos

De Orlay:

A mis padres Margot y Eliodoro que son todo para mí, y confiaron en todo momento que podía lograrlo, a ellos que me educaron y me dieron lo mejor para ayudarme a convertirme en la mujer que soy hoy.

A mis abuelos Marta, Enrique y Edelmira por todo su cariño, consejos y enseñanzas que las llevaré conmigo.

A mis hermanos Eddy y Ertis por quererme mucho y apoyarme en cada una de mis acciones.

A mi gran y único amor Ovidio que ha sido lo mejor que me ha pasado en los 5 años de la carrera, que siempre ha estado conmigo en los momentos más difíciles, a ti que me has enseñado tanto en la vida.

A mi familia que mencionarlos sería llenar varias hojas de agradecimientos. En especial a mis tíos Mirka, Idalmi,

Agradecimientos

Sandra, Eugenia, Diannis y Enriquito por estar pendiente de mí, a mis primos Marta Celia, Nenuco y Keyla.

A mi otra familia, la de mi novio que han sido muy buenos conmigo y me han apoyado en todo momento, en especial a mi abuelita Fita, a mi suegra que ha sido como una madre para mí todos estos años, por su amor y comprensión, a mi otra hermana Liannet, a mi suegro Abelardo y su esposa y a Mercedes.

A mis amistades en especial a Marisol, Pilar Isisley, Idelena, Rosana, Malidia y Lianne, que han compartido los buenos y malos momentos conmigo.

A mi compañero de tesis Osmar por sobrellevar mi exigencia.

A los tutores Rosalba, Gustavo y Rafael por dedicarnos parte de su tiempo para que este trabajo alcanzara la mejor calidad posible, a ellos muchas gracias.

A los profesores que contribuyeron en mi formación profesional.

Agradecimientos

De Osmar:

Agradezco a mis padres Nidia y Osmel y a mi hermana Naila por haberme apoyado durante estos cinco años, por confiar en mí, por brindarme todo su apoyo, cariño y amor.

A mi novia Liliana por su gran apoyo, cariño y amor incondicional durante estos cinco años.

A mi tía Gladis que ha sido como una segunda madre para mí y por apoyarme en esta etapa muy importante de mi vida, a mi abuela Cecilia por todo lo que ha hecho por mí, por ser la persona que me impulsa a seguir adelante, y por quererme como el hijo que nunca tuvo, a mi tía Dalmis, mis tíos Yoel y Raúl, mi abuela Sonia, mi abuelo Osmel, mi hermano Osmel y mis primos Aglais, Daimet, Yoelito, Randi y en especial a Rafael Rodríguez Puente por apoyarme, por quererme, por sus consejos tan importantes y por siempre preocuparse por todo lo que me acontece.

Agradecimientos

A mi familia en general que de una forma u otra han hecho posible este momento tan especial.

A mis compañeros del aula con los que he compartido estos cinco años, a mis compañeros de apartamento, a Orlando Grabiél y Eddy que más que amigos han sido como hermanos para mí y a mi compañero y amigo de cuarto Elías, con los cuales he compartido excelentes momentos.

A mis amigos que me han apoyado desde el principio, Daneysi, Elieyis, Alberto, Dennis, Rolando, Yunior, Pozo, Lester y en especial a Yoanni quien me ayudó muchísimo durante estos cinco años.

A mi compañera de tesis Orlay, por su gran dedicación, empeño y exigencia durante este trabajo.

A mis tutores por dedicarnos parte de su tiempo para que este trabajo saliera con la mejor calidad posible.

Agradecimientos

A todas las personas que de una forma u otra contribuyeron en mi formación profesional.

El departamento de opinión pública del Partido Comunista de Cuba (PCC) de la provincia de Cienfuegos, realiza la codificación y análisis de las opiniones recogidas por los activistas de los municipios en la herramienta Excel, lo cual lo convierte en un proceso complejo, debido a la cantidad de información que se maneja. En la presente investigación se desarrolla un sistema de gestión de información para facilitar dicho proceso. El mismo se desarrolló utilizando la metodología Programación Extrema (XP). Como lenguaje de programación tanto del lado del cliente como del servidor se utilizó JavaScript en su versión 1.8.5. Se utilizó NetBeans como Entorno de Desarrollo Integrado (IDE por sus siglas en inglés), a través del marco de trabajo Express. Como gestor de base de datos SQLite en su versión 3.0. Se desarrollaron las funcionalidades correspondientes a los requisitos funcionales definidos en la investigación. Se realizaron las pruebas de software a la propuesta de solución, contribuyendo a la correcta implementación de las funcionalidades y a la satisfacción por parte del cliente.

Palabras claves: análisis, codificación, gestión de información.

Contenido

Introducción	1
Capítulo 1: Fundamentación Teórica	5
1.1 Gestión de la Información.....	5
1.1.1 Sistema de Gestión de la Información.....	6
1.2 Análisis de soluciones existentes	6
1.2.1 Análisis y Diseño de una propuesta de aplicación Web para la gestión de opiniones y generación de boletines	6
1.2.2 Epi Info	6
1.2.3 Sistema de Procesamiento de Opiniones	7
1.2.4 Valoración de los sistemas similares.....	7
1.3 Herramientas y tecnologías a utilizar	8
1.3.1 Metodología de Desarrollo	8
1.3.1.1 Programación Extrema.....	8
1.3.1.2 El Proceso Unificado de Rational	9
1.3.1.3 Selección de la metodología a utilizar.....	10
1.3.2 Herramientas CASE para el modelado UML	11
1.3.2.1 Visual Paradigm for UML 8.0.....	11
1.3.3 Lenguajes de Programación	11
1.3.4 Selección del lenguaje de programación.....	13
1.3.5 Tecnologías Actuales de JavaScript	13
1.3.5.1 Node.js 0.10.35	13
1.3.5.2 Express.js 4.0.....	14
1.3.5.3 Valoración de las tecnologías actuales de JavaScript	14
1.3.6 Entorno de desarrollo integrado (IDE).....	14
1.3.6.1 NetBeans IDE 8.0.....	15
1.3.6.2 Valoración del NetBeans IDE	15
1.3.7 Sistema Gestor de Base de Datos	15
1.3.7.1 Selección del Sistema Gestor de Bases de Datos.....	17
Conclusiones del capítulo.....	17
Capítulo 2: Propuesta de solución	19

2.1 Roles del sistema	19
2.2 Listas de reservas del producto.....	20
2.2.1 Requisitos Funcionales	20
2.2.2 Aspectos no funcionales	21
2.3 Modelo conceptual	22
2.4 Historia de Usuarios	24
2.5 Estimación de esfuerzo por HU	28
2.6 Planificación del proyecto.....	29
2.5.1 Iteraciones	29
2.5.2 Plan de entregas.....	30
2.6 Diseño de la aplicación.....	31
2.6.1 Tarjetas Clase-Responsabilidades-Colaboradores	31
2.6.2 Propuesta de arquitectura.....	36
2.6.3 Modelo de datos	37
2.6.4 Patrones de diseño	38
2.6.4.1 Patrones GRASP	38
2.7 Interfaz de usuario.....	39
Conclusiones del capítulo.....	41
Capítulo 3: Implementación y prueba.....	42
3.1 Implementación	42
3.1.1 Estándares de codificación	42
3.1.2 Tareas de Ingeniería	44
3.1.3 Diagrama de despliegue	47
3.2 Análisis del funcionamiento del sistema	48
3.2.1 Validación de la entrada de datos	48
3.2.2 Mensajes del sistema.....	49
3.3 Estrategias de pruebas aplicadas al sistema.....	51
3.3.1 Tipos de prueba	51
3.3.2 Niveles de prueba	52
3.3.3 Ambiente de prueba.....	59
3.3.4 Análisis de los resultados de las pruebas.....	59
Conclusiones del capítulo.....	60

Conclusiones generales:.....	61
Recomendaciones	62
Referencias	63
Anexos	66
Anexo 1 Historia de Usuarios	66
Anexo 2 Tarjetas CRC	68
Anexo 3 Interfaz de Usuario	69
Anexo 4 Tareas de ingenierías.....	72
Anexo 5 Casos de pruebas de Aceptación.....	77

Índice de tablas

Tabla 1 Roles del sistema.....	19
Tabla 2 HU Gestionar usuario.....	25
Tabla 3 HU Gestionar opinión.....	26
Tabla 4 HU Gestionar boletín.....	26
Tabla 5 HU Gestionar activista.....	27
Tabla 6 HU Exportar boletín.....	27
Tabla 7 HU Importar boletín.....	27
Tabla 8 HU Crear opinión tipo.....	28
Tabla 9 Estimación de esfuerzo por HU.....	28
Tabla 10 Plan de iteraciones.....	29
Tabla 11 Plan de entregas.....	30
Tabla 12 Tarjeta CRC Gestionar usuario.....	31
Tabla 13 Tarjeta CRC Gestionar opinión.....	33
Tabla 14 Tarjeta CRC Gestionar boletín.....	34
Tabla 15 Tarjeta CRC Gestionar activista.....	35
Tabla 16 Tarjeta CRC Exportar boletín.....	35
Tabla 17 Tarjeta CRC Importar archivo.....	35
Tabla 18 TI Registrar usuario.....	45
Tabla 19 TI Mostrar usuarios.....	45
Tabla 20 TI Registrar opinión.....	45
Tabla 21 TI Mostrar opinión.....	45
Tabla 22 TI Registrar boletín.....	46
Tabla 23 TI Mostrar boletines.....	46
Tabla 24 TI Registrar activista.....	46
Tabla 25 TI Mostrar activista.....	46
Tabla 26 TI Exportar boletín.....	47
Tabla 27 TI Importar archivo.....	47
Tabla 28 CP de Aceptación Registrar usuario.....	55
Tabla 29 CP Aceptación Autenticar usuario.....	55
Tabla 30 CP de Aceptación Registrar opinión.....	56
Tabla 31 CP de Aceptación Mostrar opinión.....	56
Tabla 32 CP de Aceptación Registrar boletín.....	57
Tabla 33 CP de Aceptación Mostrar boletín.....	57
Tabla 34 CP de Aceptación Registrar activista.....	57
Tabla 35 CP de Aceptación Mostrar activista.....	58
Tabla 36 CP de Aceptación Exportar boletín.....	58
Tabla 37 CP de Aceptación Importar archivo.....	58
Tabla 38 CP de Aceptación Crear opinión tipo.....	59
Tabla 39 Ambiente de prueba de la propuesta de solución.....	59

Índice de tablas

Tabla 40 HU Evaluar activista	66
Tabla 41 HU Mostrar evaluaciones	66
Tabla 42 HU Asignar opinión tipo	67
Tabla 43 HU Eliminar opinión tipo	67
Tabla 44 HU Mostrar opinión tipo	67
Tabla 45 HU Mostrar cantidad de opiniones.....	68
Tabla 46 Tarjeta CRC Gestionar activista	68
Tabla 47 Tarjeta CRC Evaluar activista.....	68
Tabla 48 Tarjeta CRC Insertar opinión tipo.....	68
Tabla 49 Tarjeta CRC Asignar opinión tipo.....	69
Tabla 50 Tarjeta CRC Eliminar opinión tipo.....	69
Tabla 51 Tarjeta CRC Mostrar cantidad de opiniones	69
Tabla 52 Tarjeta CRC Mostrar opinión tipo.....	69
Tabla 53 TI Autenticar usuario	73
Tabla 54 TI Eliminar usuario.....	73
Tabla 55 TI Editar usuario.....	73
Tabla 56 TI Cambiar contraseña.....	73
Tabla 57 TI Cerrar sesión.....	74
Tabla 58 TI Editar opinión	74
Tabla 59 TI Eliminar opinión	74
Tabla 60 TI Editar boletín	74
Tabla 61 TI Eliminar boletín	75
Tabla 62 TI Eliminar activista.....	75
Tabla 63 TI Editar activista.....	75
Tabla 64 TI Evaluar activista	75
Tabla 65 TI Mostrar evaluaciones.....	76
Tabla 66 TI Crear opinión tipo.....	76
Tabla 67 TI Asignar opinión tipo	76
Tabla 68 TI Eliminar opinión tipo.....	76
Tabla 69 TI Mostrar opiniones tipo	77
Tabla 70 TI Mostrar cantidad de opiniones.....	77
Tabla 71 CP de Aceptación Mostrar usuarios	77
Tabla 72 CP de Aceptación Cambiar contraseña	78
Tabla 73 CP de Aceptación Editar usuario	78
Tabla 74 CP de Aceptación Eliminar opinión.....	78
Tabla 75 CP de Aceptación Cerrar sesión	79
Tabla 76 CP de Aceptación Editar datos de las opiniones.....	79
Tabla 77 CP de Aceptación Eliminar opinión.....	79
Tabla 78 CP de Aceptación Editar boletín.....	80
Tabla 79 CP de Aceptación Eliminar boletín.....	80

Índice de tablas

Tabla 80 CP de Aceptación Eliminar activista	80
Tabla 81 CP de Aceptación Editar activista	81
Tabla 82 CP de Aceptación Evaluar activista	81
Tabla 83 CP de Aceptación de Mostrar evaluaciones.....	81
Tabla 84 CP de Aceptación de Asignar opinión tipo.....	82
Tabla 85 CP de Aceptación Eliminar opinión tipo	82
Tabla 86 CP de Aceptación Mostrar cantidad de opiniones.....	82

Índice de ilustraciones

Figura 1 Fases y Flujos de RUP	10
Figura 2 Diagrama conceptual del negocio	23
Figura 3 Representación del patrón arquitectónico MVC	37
Figura 4 Modelo físico de la base de datos de la propuesta de solución.....	37
Figura 5 Interfaz de usuario Registrar usuario	40
Figura 6 Interfaz de usuario Mostrar opinión	40
Figura 7 Interfaz de usuario Mostrar boletín.....	40
Figura 8 Interfaz de usuario Exportar boletín	41
Figura 9 Interfaz de usuario Importar archivo.....	41
Figura 10 Estándar de codificación CamelCase.....	43
Figura 11 Comentarios de una única línea.....	43
Figura 12 Comentarios en varias líneas.....	44
Figura 13 Diagrama de despliegue	48
Figura 14 Mensaje de información Cambiar Contraseña	49
Figura 15 Mensaje de error Registrar usuario	50
Figura 16 Mensaje de confirmación Eliminar boletín	51
Figura 17 Prueba unitaria de la clase Usuario	54
Figura 18 Prueba unitaria de las clases Opinión y Boletín	54
Figura 19 Resultados de las pruebas.....	60
Figura 20 Interfaz de usuario Mostrar usuarios.....	69
Figura 21 Interfaz de usuario Registrar opinión.....	70
Figura 22 interfaz de usuario Registrar boletín.....	70
Figura 23 Interfaz de usuario Registrar activista	71
Figura 24 Interfaz de usuario Mostrar activistas.....	71
Figura 25 Interfaz de usuario Cambiar contraseña	72
Figura 26 Interfaz de usuario Opciones de filtrado.....	72
Figura 27 Interfaz de usuario Mostrar evaluaciones de los activistas.....	72
Figura 28 Interfaz de usuario Mostrar opiniones tipos.....	72

Introducción

La Revolución cubana, desde sus inicios, creó sus propios instrumentos para sistematizar el conocimiento de la opinión pública y sumarlo a la constante atención de los dirigentes e instituciones políticas al sentir del pueblo.

Desde los primeros años de la Revolución, el Sistema de Opinión del Pueblo contaba con un mínimo de personal profesional en municipios, provincias y el Comité Central del PCC, su fundamento y sostén ha sido la abnegada labor de miles de militantes que de forma voluntaria, han cumplido la tarea de transmitir a todas las instancias del PCC y de otras instituciones del país, los puntos de vista de la sociedad.

En 1991, por decisión del Buró Político, se crea el Centro de Estudios Sociopolíticos y de Opinión (CESPO) y en 1998, se hizo una importante reestructuración dirigida a fortalecer el movimiento de activistas y los equipos provinciales, adoptando la forma actual.

El CESPO tiene la función de dirigir metodológicamente el Sistema de Opinión del Pueblo, además de conocer y estudiar los estados de opinión que se producen como resultado de acontecimientos cotidianos o extraordinarios, de carácter nacional o internacional, y realizar investigaciones y estudios sociopolíticos de interés para el Comité Central del Partido.

El Equipo Provincial de Estudios Sociopolíticos y de Opinión (EESPO) es la estructura que posibilita al CESPO cumplir con la misión que la máxima dirección del Partido le ha encomendado. Organiza, orienta, coordina y controla el trabajo de los equipos de estudios sociopolíticos y de opinión en cada uno de los municipios del territorio. Estudia los estados de opinión que se originan en la provincia como resultado de acontecimientos cotidianos o extraordinarios, de carácter nacional o internacional y realiza investigaciones y estudios sociopolíticos de interés para el Buró Provincial del Partido. Contando para ello con un grupo de activistas distribuidos por toda la provincia encargados de hacer llegar sus boletas al municipio, al menos cuatro días diferentes de la semana. Este es uno de los indicadores que se tienen en cuenta para evaluar al activista como expresión de su sistematicidad en el trabajo.

Las opiniones sobre las temáticas de la vida cotidiana recogidas en la semana, de lunes a domingo, deben entregarse en la misma semana, en el municipio. Existen opiniones que de acuerdo a su prioridad, por estar relacionadas con algún tema importante de la actualidad, o por constituir un asunto de connotación pública

Introducción

o un rumor, el activista debe darle carácter de entrega rápida a la boleta, por lo que debe ser ágil en tal sentido y hacerla llegar al municipio, el mismo día en que se recepciona. En esta situación, puede utilizar de forma excepcional el teléfono, entregando con posterioridad la boleta, en la que debe consignar que ya esa opinión se informó por vía telefónica.

En cada municipio existe un técnico que codifica dichas opiniones haciendo uso de la herramienta Excel como sistema de gestión y análisis de la información, este proceso no es más que la asignación a la opinión de un código compuesto por tema, subtema, asunto, subasunto, sentido, fecha, red, activista, provincia, municipio, consejo popular y el tipo de boletín, dichos parámetros son en los que luego los funcionarios se apoyan para realizar los análisis y reportes a las instancias superiores. Luego de que se realice la codificación de las opiniones se envía un parte con las opiniones del día al EESPO, donde existen un grupo de especialistas encargados de revisar la codificación de las opiniones, enviándose para el CESPO en la provincia La Habana, un boletín con las opiniones más importantes del día.

A finales de cada mes se realiza un análisis de contenido, en el que se hace un resumen de las opiniones del pueblo por cada tema de interés social, del cual se sacan las opiniones más relevantes (las que más se repiten o contengan alguna importancia de análisis en sentido social). Este proceso genera un gran cúmulo de documentos e informes de forma manual, difícil de gestionar en la herramienta Excel, ya que brinda facilidades estadísticas, pero no ofrece opciones para hacer un análisis exhaustivo y específico de las opiniones de forma eficiente, debido a que no se puede guardar la información de forma automática y guardar de manera reiterada demora el proceso, no se obtienen todas las tablas estadísticas que modelan las actividades del negocio, provocándoles atraso en la entrega de la información recibida por los municipios, además el departamento cuenta con pocas estaciones de trabajo, las cuales presentan las siguientes características: 256 MB RAM, 40 GB de disco duro y procesadores Intel Pentium y son compartidas entre varios compañeros.

Teniendo en cuenta la situación anterior, se define como **problema a resolver** el proceso de gestión de las opiniones públicas en el PCC de Cienfuegos dificulta la codificación y análisis de la misma.

El **objeto de la investigación** se centra en los sistemas de gestión de información para el análisis de opiniones públicas. En el que se especifica como **campo de acción** los procesos de codificación y análisis de opiniones públicas.

Para dar solución al problema anterior se establece como **objetivo general**: desarrollar un sistema de

Introducción

gestión de información que mejore el proceso de la codificación y análisis de las opiniones públicas en el PCC de Cienfuegos.

Los **objetivos específicos** son:

- Establecer los referentes teóricos y metodológicos sobre el proceso de la gestión de información de opinión pública a través de soluciones informáticas.
- Caracterizar el estado actual de los procesos de gestión de información de opinión pública del PCC de Cienfuegos.
- Desarrollar un sistema informático que permita la mejora del proceso de gestión de la información.
- Realizar pruebas al Sistema de Gestión de Información en correspondencia con el objetivo de verificar su utilidad con respecto a las opiniones del pueblo.

Se tiene como **Hipótesis** que si se desarrolla un sistema de gestión de información que permita la codificación y análisis de las opiniones públicas en el PCC de Cienfuegos entonces se mejorará la gestión de dicho proceso.

Para dar cumplimiento a los objetivos específicos se planifican las siguientes **tareas de la investigación**:

- Revisión bibliográfica para conformar el estado del arte de la investigación.
- Revisar la existencia de otras aplicaciones o soluciones similares.
- Selección de la metodología a utilizar en el desarrollo del sistema.
- Estudio y selección de las herramientas y tecnologías a utilizar para la implementación del sistema.
- Identificación de los requerimientos funcionales y no funcionales del sistema de gestión de información que se obtiene en el departamento de opinión pública del PCC de Cienfuegos.
- Desarrollo de los artefactos de acuerdo a la metodología de desarrollo seleccionada.
- Implementación de la propuesta de solución.
- Realización de las pruebas de software a la propuesta de solución.

Los **métodos teóricos** utilizados en la investigación son los siguientes:

Analítico-Sintético: Con este método se analizaron todas las informaciones obtenidas relacionadas con el tema, que posteriormente se sintetizaron e integraron en la confección de la solución propuesta.

Introducción

Histórico-Lógico: Se utilizó con el objetivo de verificar teóricamente cómo ha evolucionado el tema que se está tratando en esta investigación, además, de evaluar el desarrollo de sistemas similares, con el fin de que puedan ser usadas para la elaboración de la solución propuesta.

Modelado: Este método se utilizó para modelar todos los artefactos que se generan a lo largo de todo el proceso de desarrollo, con el objetivo de propiciar un entendimiento real de la solución.

Para un mejor entendimiento del trabajo de diploma, el mismo se organizó en tres capítulos. A continuación se describen las diferentes temáticas que se abordan en cada uno de ellos:

➤ Capítulo 1: Fundamentación teórica de la investigación

Este capítulo abarca todos los conceptos que permiten un mejor entendimiento de los Sistemas de Gestión de Información. En él se realiza un estudio de sistemas y soluciones similares, se describen las tecnologías y las herramientas que se emplearán para dar cumplimiento al objetivo general de la investigación, así como las metodologías utilizadas para el desarrollo de un sistema de gestión de información.

➤ Capítulo 2: Propuesta de solución

En el capítulo se describe la propuesta de solución, haciendo uso de los artefactos que genera la metodología de desarrollo utilizada (XP). Se especifican los requisitos definidos obteniendo como resultado el diseño y la estructuración de la propuesta de solución.

➤ Capítulo 3: Implementación y prueba

Recoge lo referido a la implementación del sistema y sus funcionalidades, se realiza la validación del mismo mediante las pruebas de software, para el aseguramiento de la calidad.

Capítulo 1: Fundamentación Teórica

Capítulo 1: Fundamentación Teórica

En la actualidad el avance de las tecnologías han propiciado a la humanidad numerosos cambios científicos y tecnológicos con el uso de programas informáticos, los cuales con el paso del tiempo han progresado en el uso de sitios web, como sistemas de gestión de información, herramientas de apoyo a la toma de decisiones en las organizaciones.

En este capítulo se hace un estudio de las herramientas y las metodologías utilizadas para poder realizar la propuesta de solución. Se realiza una investigación de los sistemas informáticos que se usan en la actualidad para dar solución a problemas similares, además de los que se han desarrollado en la Universidad de las Ciencias Informáticas (UCI). Se tiene en cuenta los lenguajes de programación web, los Sistemas Gestores de Base de Datos (SGBD) más utilizados, así como programas que faciliten el trabajo con estas herramientas. Finalmente, se brinda una propuesta de las herramientas que se utilizarían para llevar a cabo la presente investigación.

1.1 Gestión de la Información

La gestión de la información es el proceso de organizar, evaluar, presentar, comparar los datos en un determinado contexto, controlando su calidad, de manera que esta sea veraz, oportuna, significativa, exacta y útil y que esta información esté disponible en el momento que se le necesite. La misma se encamina al manejo de la información, documentos, metodologías, informes, publicaciones, soportes y flujos en función de los objetivos estratégicos de una organización (Quiroga, 2002).

La finalidad de la gestión de la información es ofrecer mecanismos que permiten a la organización adquirir, producir y transmitir, al menor coste posible, datos e informaciones con una calidad, exactitud y actualidad suficientes para servir a los objetivos de la organización. En términos perfectamente entendibles sería conseguir la información adecuada, para la persona que lo necesita, en el momento que lo necesita, al mejor precio posible para tomar la mejor de las decisiones (Arévalo, 2007).

Se puede concluir que la gestión de la información es un proceso mediante el cual se planifican, organizan, dirigen y controlan los recursos de información, para ser usada en el desarrollo y el éxito de una organización.

Capítulo 1: Fundamentación Teórica

1.1.1 Sistema de Gestión de la Información

Según el concepto que establece la Universidad de California un Sistema de Gestión de Información (IMS por sus siglas en inglés) es la aplicación de la tecnología de la información para apoyar las principales funciones y actividades de un negocio ya sea del sector privado o institución del sector público. En el pasado, las organizaciones reconocían la importancia de la gestión de los recursos, como el trabajo, el capital y las materias primas. Hoy en día, está ampliamente aceptado que la gestión del recurso de información es igualmente importante. Un IMS apoya el proceso de recolección, manipulación, almacenamiento, distribución y utilización de los recursos de información de una organización (University of California, 2013).

A partir de lo planteado se define que los Sistemas de Gestión de Información, son sistemas computarizados que almacenan los datos e informaciones de una empresa u organización, como apoyo a la gestión y toma de decisiones en dichas organizaciones.

1.2 Análisis de soluciones existentes

Hoy en día el desarrollo de las TIC ha posibilitado gran avance en el uso de técnicas y tecnologías, que permiten el análisis de la información con la calidad requerida. Luego del estudio realizado, se identificó que actualmente existen sitios web dedicados a la gestión de opiniones de una empresa u organización, las cuales se describen a continuación:

1.2.1 Análisis y Diseño de una propuesta de aplicación Web para la gestión de opiniones y generación de boletines

Es un sistema que se desarrolló en la UCI, con el fin de generar boletines y gestionar todas las opiniones espontáneas que se procesan en el CESPO y en el EESPO. Dentro de sus funcionalidades permite gestionar un tema mensual, cargar opiniones y buscar opiniones (Olazabal, 2009).

1.2.2 Epi Info

Es un sistema estadístico creado inicialmente para su utilización en epidemiología, utilizado en la mayoría de las provincias, con el objetivo de tener un control de las opiniones en cuanto a (tema, subtema, municipio, activista, etc.), está compuesto por varios módulos entre los que se encuentran:

Capítulo 1: Fundamentación Teórica

- EPED (Editor de textos): Es un procesador de textos para crear cuestionarios de tipo Epi Info o procesar textos.
- ENTER (Base de datos): Permite introducir y editar datos, además de revisar su estructura. Se crean los ficheros de datos .REC para que luego sean analizados.
- ANALYSIS (Analiza datos): Permite manipular y analizar registros, hacer listados, tablas y gráficas estadísticas con ficheros de datos.
- CHECK (Valida entradas): Permite establecer rangos, patrones de salto, codificación. Se pueden ejecutar operaciones matemáticas y lógicas al introducir datos y relacionar campos para rutinas escritas en otros programas informáticos.
- CONVERT (Exporta datos): Es el método de exportar desde un fichero de datos Epi Info a formatos de otros programas estadísticos.
- IMPORT (Importa datos): Permite importar ficheros creados en otros sistemas.
- MERGE (Une ficheros): Permite combinar de diferentes maneras ficheros de datos.

El programa Epi Info es una aplicación de consola que solo puede ser usada en el sistema operativo (SO) propietario Windows, lo que implica que si la institución decide migrar hacia software libre sería imposible su utilización, además de que no presenta una interfaz de usuario amigable (Melissa Alperin, Katheen R. Miner, 1997).

1.2.3 Sistema de Procesamiento de Opiniones

Es una aplicación web que permite gestionar las opiniones emitidas por estudiantes, profesores y trabajadores de la UCI, la cual cuenta con funcionalidades como revisión de las opiniones, actualización del estado de las opiniones, reportes estadísticos y publicación simultánea de varios temas de debates (Dashliell García Martínez, 2011).

1.2.4 Valoración de los sistemas similares

Luego del estudio de las herramientas anteriores, se determinó que estos sistemas no satisfacen las necesidades de gestión de las opiniones del PCC, debido a que fueron desarrollados a la medida para dar solución a problemas específicos. Sin embargo, se tienen en cuenta sus características para el desarrollo de la investigación, tales como:

- Un módulo de administración encargado de la gestión de usuarios, roles y permisos.
- Generación de reportes de la información para visualizar los datos más importantes.
- Generación de evaluaciones de sus trabajadores.
- Registro y control de la información recogida.

Capítulo 1: Fundamentación Teórica

1.3 Herramientas y tecnologías a utilizar

Para el desarrollo de la solución es necesario el estudio de herramientas y tecnologías que guíen el proceso de desarrollo, a continuación se presentan descripciones de las mismas.

1.3.1 Metodología de Desarrollo

En el desarrollo de software es necesario un proceso que nos ayude a estructurar, planificar y gestionar las tareas y actividades que se deben realizar a lo largo del proceso de desarrollo. Las metodologías de desarrollo proporcionan un conjunto de técnicas para lograr la calidad y el éxito de un software, diferenciándose en dos grandes grupos, ágiles y pesadas.

Las denominadas metodologías tradicionales o pesadas, son aquellas que se caracterizan por un fuerte énfasis en la planificación y control del proyecto, y en la especificación precisa de requisitos y modelado. Ello impone una disciplina de trabajo sobre el proceso de desarrollo de software con un excesivo carácter formal (Paloma, 2001). Las mismas han demostrado ser efectivas y necesarias en proyectos de gran tamaño (en cuanto a tiempo y recursos). Por su parte las metodologías clasificadas como ágiles son aquellas que dan mayor valor al individuo, a la colaboración estrecha y directa con el cliente, y al desarrollo incremental del producto con iteraciones muy cortas. Este enfoque ha mostrado ser efectivo en entornos cambiantes, donde los requisitos no están bien definidos o es muy probable que sufran algún cambio. Son muy útiles en el desarrollo de proyectos que cuentan con cortos períodos de tiempo para su ejecución, sin dejar a un lado la calidad (Patricio Letelier, 2006).

Dentro de las metodologías se destacan Programación Extrema (XP, por sus siglas en inglés) y el Proceso Unificado de Rational RUP, debido a que son las más utilizadas y documentadas en la UCI. Además por el poco tiempo que resta para el desarrollo de la solución. A continuación se muestran algunas de las características de estas, de forma breve y se fundamenta el porqué de la selección de XP como guía del proceso de desarrollo de software para la presente investigación.

1.3.1.1 Programación Extrema

La programación extrema (XP por sus siglas en inglés) es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado.

Capítulo 1: Fundamentación Teórica

Se caracteriza por ser una metodología ágil, la cual ofrece al cliente el software que necesita y cuando lo necesita, además de combinarse en ella las mejoras prácticas para el desarrollo de software. Es uno de los procesos ágiles más destacados, diferenciándose de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Tiene como meta la disminución de costes y su objetivo principal es satisfacer a los clientes y potenciar al máximo el trabajo en grupo, además los jefes de proyecto, los desarrolladores y el cliente son parte del equipo de desarrollo. En el caso de los desarrolladores deben conocer que durante el ciclo de desarrollo de un proyecto de software los cambios aparecen, cambian los requisitos, las reglas del negocio, el personal y la tecnología, por lo tanto este no es el principal problema ya que va a suceder, en lo que reside el principal problema es en la capacidad que tengan para adaptarse a esos cambios (Letelier, P. y Penadés, M, Valencia, 2008).

Promueve el trabajo en equipo, preocupándose en todo momento del aprendizaje de los desarrolladores y estableciendo un buen clima de trabajo. Este tipo de método se basa en una realimentación continuada entre el cliente y el equipo de desarrollo con una comunicación fluida entre todos los participantes, también busca simplificar las soluciones implementadas y coraje para los múltiples cambios (Extreme Programming, 2009).

1.3.1.2 El Proceso Unificado de Rational

El Proceso Unificado Rational (RUP por sus siglas en inglés) es un marco de trabajo del proceso de desarrollo de software. RUP no es un proceso preceptivo concreto individual, sino un marco de trabajo de proceso adaptable, con la idea de ser adaptado por las organizaciones de desarrollo y los equipos de proyecto de software que seleccionarán los elementos del proceso que sean apropiados para sus necesidades. Incluye una base de conocimientos con artefactos de ejemplos y descripciones detalladas para muchos tipos diferentes de actividades.

RUP se basa en un conjunto de módulos o elementos de contenido, que describen qué se va a producir, las habilidades necesarias requeridas y la explicación paso a paso describiendo cómo se consiguen los objetivos de desarrollo.

Los módulos principales, o elementos de contenido engloban las fases del ciclo de vida del proyecto. RUP determina que el ciclo de vida del proyecto consiste en cuatro fases que permiten que el proceso sea presentado a alto nivel.

Capítulo 1: Fundamentación Teórica

Además, se divide en nueve flujos de trabajo. (Modelado del Negocio, Requisitos, Análisis y Diseño, Implementación, Pruebas, Despliegue, Gestión de Cambio y Configuraciones, Gestión del Proyecto y Entorno) (Ivar Jacobson, 2000).

La figura 1 modela el flujo de trabajo de la metodología RUP:

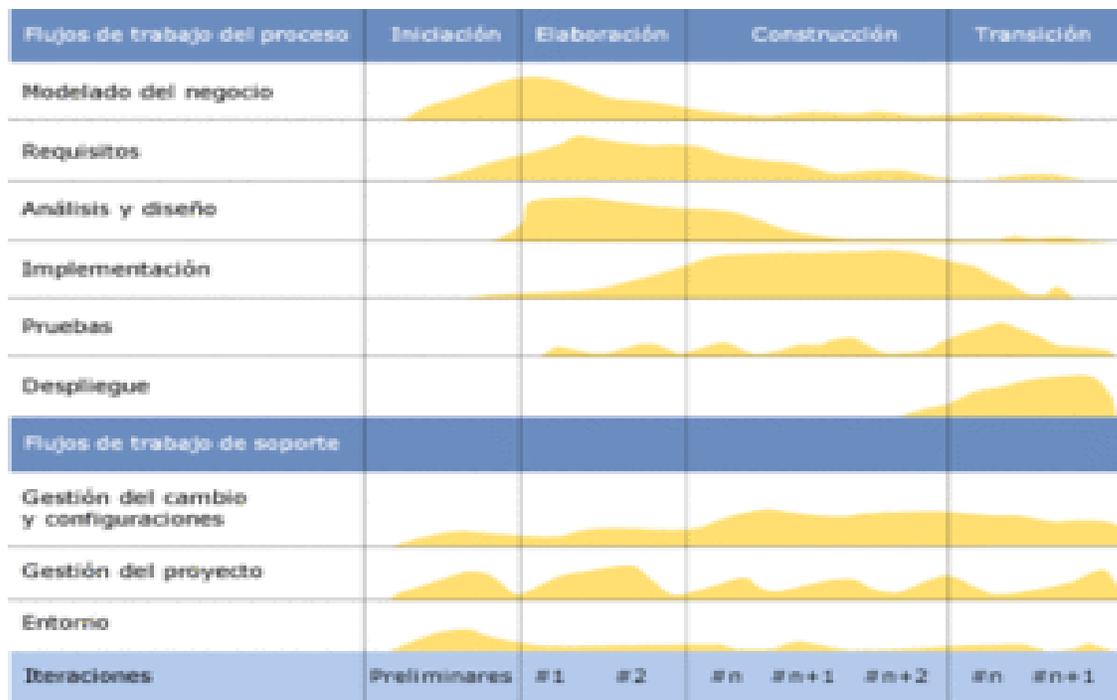


Figura 1 Fases y Flujos de RUP

1.3.1.3 Selección de la metodología a utilizar

Luego de haber realizado un estudio de las metodologías, se decide utilizar XP, debido a que se adapta a las características que posee la propuesta de solución a desarrollar, se cuenta con un corto período de tiempo para el desarrollo, los requisitos pueden sufrir cambios a lo largo del proceso, el cliente forma parte del equipo de trabajo, el cual cuenta con poco personal, la dimensión de la propuesta de solución es pequeña (primera versión) y el horario de trabajo es de un máximo de 40 horas por semana sin trabajar horas extras, permite revisar el código continuamente, así como realizar pruebas intensivas, lo que requiere de un proceso de desarrollo ágil. Otro aspecto valorado es que es una metodología de la cual los autores de la investigación tienen dominio.

Capítulo 1: Fundamentación Teórica

1.3.2 Herramientas CASE para el modelado UML

Las herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores (De la Cruz Rodríguez, 2010). La herramienta seleccionada para el modelado de la solución es Visual Paradigm for UML en su versión 8.0.

1.3.2.1 Visual Paradigm for UML 8.0

Visual Paradigm for UML: Es una herramienta para la Ingeniería de Software Asistida por Computadora que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todas las etapas del ciclo de vida de desarrollo de un software. Es un software de carácter comercial, con una versión libre para la comunidad (Visual Paradigm for UML Community Edition (VP-UML CE)) y el resto con versiones de evaluación (Paradigm, Visual, 2014).

El uso de la herramienta Visual Paradigm permitirá el modelado del sistema con el fin de obtener un producto de forma rápida y con excelente calidad. Posee una buena documentación, tutoriales de UML y demostraciones interactivas, además, el equipo de trabajo ya cuenta con experiencia en el trabajo con dicha herramienta, pues se ha usado en anteriores investigaciones.

1.3.3 Lenguajes de Programación

Actualmente existen diferentes lenguajes de programación para desarrollar aplicaciones web, a continuación se analizan PHP y JavaScript que son los lenguajes más utilizados para desarrollar aplicaciones web. Además, se utiliza HTML para el diseño de las plantillas web.

PHP 5.3.8

PHP es un lenguaje de script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que

Capítulo 1: Fundamentación Teórica

contiene este código) para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML. Debido a esto y a que es de código abierto es uno de los más populares y extendido en la Web. PHP fue desarrollado originalmente por Rasmus Lerdorf en 1994 pero como está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores (Welling, 2005).

Desventaja de PHP:

A pesar de que PHP es gratuito y está bien documentado, es un lenguaje que presenta desventajas que hacen que el equipo de desarrollo seleccione a JavaScript. Ya que PHP es un lenguaje interpretado y más lento que otros. Al conectarse a la base de datos y leer los datos del formulario necesita de un alto rendimiento, por lo que consume más recurso y tarda más tiempo en mostrar una página web.

JavaScript 1.8.5

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como aparición y desaparición de texto, animaciones, acciones que se activan al pulsar botones u otros elementos y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Eguíluz Pérez, Javier, 2009). La selección de JavaScript como el lenguaje de script del lado del cliente, está determinada por el hecho de usar la biblioteca jQuery, la cual está escrita en el mencionado lenguaje.

HTML 5

El Lenguaje de Etiquetado de Hipertexto es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado W3C (World Wide Web Consortium). Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de la misma manera en cualquier navegador de cualquier sistema operativo. La W3C define el lenguaje HTML como “Un lenguaje comúnmente utilizado para la publicación de hipertexto en la Web y desarrollado con la idea de que cualquier persona o tipo de dispositivo pueda acceder a

Capítulo 1: Fundamentación Teórica

la información en la Web”. HTML utiliza etiquetas que marcan elementos y estructuran el texto de un documento (W3C, 2014).

HTML5 propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. HTML5 provee los elementos estructurales, CSS se encarga de volver esa estructura utilizable y atractiva a la vista, y JavaScript tendrá el poder necesario para proveer dinamismo y construir aplicaciones web completamente funcionales y con mayor capacidad de interacción con el usuario (Gauchat, 2012).

1.3.4 Selección del lenguaje de programación

Por lo anteriormente analizado se decide utilizar JavaScript como lenguaje de programación, debido a que es muy utilizado para crear páginas web dinámicas. JavaScript ofrece como ventaja la posibilidad de compilar los programas, sin la necesidad de ejecutarlos, en otras palabras, los programas escritos en JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. Además, el equipo de trabajo cuenta con buena experiencia en el uso de este lenguaje de programación debido a que ha sido usado en proyectos anteriores.

1.3.5 Tecnologías Actuales de JavaScript

Los nuevos cambios existentes en la humanidad hacen que los ciudadanos tengan que actualizarse constantemente en conocimientos relacionados con las nuevas tecnologías. Las nuevas tecnologías, relacionadas con la informática y la informatización de la sociedad, agilizan, optimizan y perfeccionan algunas actividades que son de interés para el desarrollo de una sociedad, dentro de ellas se destacan las tecnologías actuales de JavaScript como: Node.js, y Express. A continuación se describen cada una de ellas.

1.3.5.1 Node.js 0.10.35

Node.js es un entorno de programación en JavaScript para el Backend basado en el motor V8 de JavaScript del navegador Google Chrome y orientado a eventos, lo que lo hace muy rápido a la hora de crear servidores web y emplear tiempo real. Fue creado en 2009 por Ryan Dahl y aunque aún es reciente, las últimas versiones lo hacen robusto además de la comunidad de desarrolladores que posee. Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el lado del servidor. No solo se utiliza en servidor, se ha extendido tanto que se emplea en Stylus, un preprocesador CSS, y en Grunt, un gestor de tareas basado en JavaScript.

Capítulo 1: Fundamentación Teórica

Node.js incorpora varios "módulos básicos" compilados en el propio binario, como por ejemplo el módulo de red, que proporciona una capa para programación de red asíncrona y otros módulos fundamentales, como Path, FileSystem, Buffer, Timers y el de propósito más general Stream. Es posible utilizar módulos desarrollados por terceros, ya sea como archivos "node" precompilados, o como archivos en JavaScript plano. Los módulos JavaScript se implementan siguiendo la especificación CommonJS para módulos, utilizando una variable de exportación para dar a estos scripts acceso a funciones y variables implementadas por los módulos basados en JavaScript (Rolando Jesús Hidalgo, 2014).

1.3.5.2 Express.js 4.0

Es un marco de trabajo para Node.js que permite ofrecer una solución integrada que se puede implementar rápidamente. Está basado en el motor de conexión middleware. Además permite configurar el middleware para responder a solicitudes HTTP, permite realizar diferentes acciones basadas en el método HTTP y URL para definir una tabla de enrutamiento. Permite procesar dinámicamente documentos HTML basados en el suministro de argumentos a las plantillas (Teixera, 2012).

1.3.5.3 Valoración de las tecnologías actuales de JavaScript

El uso de estos marcos de trabajo es muy importante para el desarrollo de la propuesta de solución, ya que Express.js es el encargado de todo lo relacionado con el uso del protocolo HTTP y las direcciones web URL, lo que permite que el desarrollo y las pruebas sean más fáciles, utilizando a Node.js como entorno de programación que incorpora las tecnologías antes mencionadas. Además, poseen licencia (Instituto Tecnológico de Massachusetts) (MIT por sus siglas en inglés), la cual es libre.

1.3.6 Entorno de desarrollo integrado (IDE)

Un IDE es un programa informático compuesto por un conjunto de herramientas de programación. Consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Provee un marco de trabajo amigable para el lenguaje de programación correspondiente (IDE, 2009).

Capítulo 1: Fundamentación Teórica

1.3.6.1 NetBeans IDE 8.0

NetBeans IDE permite desarrollar aplicaciones web rápidas y fácilmente para escritorios Java, móviles y, así como aplicaciones HTML5, JavaScript y CSS. Es gratuito y de código abierto y tiene una gran comunidad de usuarios y desarrolladores de todo el mundo.

NetBeans es el IDE oficial de Java 8. Con sus editores, analizadores de código y convertidores, puede actualizar de forma rápida y sin problemas sus aplicaciones para utilizar las nuevas construcciones del lenguaje Java 8, como operaciones funcionales y las referencias a métodos. Analizadores de lotes y convertidores son proporcionados a buscar a través de múltiples aplicaciones al mismo tiempo, haciendo coincidir los patrones para la conversión a las nuevas construcciones del lenguaje Java 8. NetBeans IDE establece el estándar para el desarrollo de tecnologías (NetBeans, 2014).

1.3.6.2 Valoración del NetBeans IDE

El uso del NetBeans IDE es de mucha importancia para el desarrollo de la propuesta de solución, debido a que es un entorno intuitivo y posee características que agilizan el proceso de desarrollo, soporta varios componentes que lo hacen compatibles con diferentes lenguajes de programación, otras tecnologías y está liberado bajo la Licencia Pública General GNU/GPL. Además, el equipo de trabajo cuenta con experiencia en el uso del NetBeans IDE, pues el mismo ha sido utilizado en el desarrollo de proyectos anteriores.

1.3.7 Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un software cuyo objetivo es proporcionar una interfaz entre la base de datos (BD), el usuario y las aplicaciones que la utilizan. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la BD, el control de acceso, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad (Pérez Garzón, 2011).

MongoDB

MongoDB es una base de datos no relacional (NoSQL) de código abierto, orientado a documentos, que guarda los datos en ficheros de tipo JSON (JavaScript Object Notation) pero en forma binaria (BSON) para hacer la integración de una manera más rápida. Se

Capítulo 1: Fundamentación Teórica

pueden ejecutar operaciones en JavaScript en su consola en lugar de consultas SQL. Además se integra con Node.js con los driver propios.

Principales características:

MongoDB soporta la búsqueda por campos, consultas de rangos y expresiones regulares. Cualquier campo en un documento de MongoDB puede ser indexado, al igual que es posible hacer índices secundarios. Además puede ser utilizado con un sistema de archivos, tomando la ventaja de la capacidad que tiene para el balanceo de carga y la replicación de datos utilizando múltiples servidores para el almacenamiento de archivos (MongoDB, 2003).

SQLite 3

SQLite es un sistema de gestión de bases de datos relacional que incluye transacciones de bases de datos atómicas, consistencia de bases de datos, aislamiento y durabilidad (ACID, por sus siglas en inglés), contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host (SQLite, 2000).

Principales características:

SQLite 3 usa un sistema de tipos inusual. En lugar de asignar un tipo a una columna como en la mayor parte de los sistemas de bases de datos SQL, los tipos se asignan a los valores individuales. Por ejemplo, se puede insertar un String en una columna de tipo entero (a pesar de que SQLite tratará en primera instancia de convertir la cadena en un entero). Algunos usuarios consideran esto como una innovación que hace que la base de datos sea mucho más útil, sobre todo al ser utilizada desde un lenguaje de scripting de tipos dinámicos.

Capítulo 1: Fundamentación Teórica

PostgreSQL

Es un sistema de gestión de bases de datos objeto-relacional. Clasifica como software libre por su distribución bajo la licencia BSD, por lo que su código fuente y ficheros binarios pueden ser utilizados, modificados, redistribuidos y hasta incluidos en software con carácter no libre. Utiliza un modelo cliente-servidor y utiliza multiprocesos en vez de multihilos para garantizar que el sistema sea estable. Las características técnicas que posee lo hacen uno de los gestores más potentes y robustos del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo (Martinez, 2014). Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo al mismo tiempo al sistema.

1.3.7.1 Selección del Sistema Gestor de Bases de Datos

Una vez realizada la investigación de los sistemas de gestores de base de datos se decide utilizar SQLite debido a que en su versión 3, SQLite no tiene problemas de escalabilidad, presenta concurrencia, o sea que varios procesos o hilos pueden acceder a la misma base de datos sin problemas y su durabilidad y verificabilidad de los datos es permanente. Además es un SGBD que se integra muy bien con el entorno de programación Node.js.

Conclusiones del capítulo

El estudio realizado sobre la gestión de la información permitió identificar los diferentes conceptos y definiciones importantes para dar solución al problema de la investigación, así como las herramientas y tecnologías a tener en cuenta para el posterior desarrollo de la investigación.

- Se determinó que las herramientas similares no cumplen con las necesidades del problema planteado por el departamento de opinión pública del PCC de Cienfuegos.
- Como guía para el proceso de desarrollo se escogió la metodología ágil XP, para el modelado de los artefactos, la herramienta CASE Visual Paradigm en su versión 8.0. Determinándose que la propuesta de solución será una aplicación web, para su desarrollo se seleccionó como lenguaje de programación a JavaScript 1.8.5 y HTML 5, para el almacenamiento de la información al gestor de bases de datos SQLite 3, el entorno de programación Node.js en su versión 0.10.35, como framework de

Capítulo 1: Fundamentación Teórica

desarrollo a Express en su versión 4.0 y como IDE fue seleccionado el NetBeans 8.0.

Capítulo 2: Propuesta de solución

Capítulo 2: Propuesta de solución

La metodología XP define una serie de artefactos que son muy importantes a tener en cuenta para el desarrollo del sistema. En el presente capítulo se describen las fases planificación y diseño de la propuesta de solución. Se desarrollan los artefactos correspondientes a cada una de las fases antes mencionadas, como el Modelo de datos, las Historias de Usuarios, Plan de Entregas, Iteraciones y las Tarjetas CRC (Clases, Responsabilidad y Colaboración).

2.1 Roles del sistema

El sistema permitirá a los usuarios ejecutar un conjunto de funcionalidades a través de los cuales se establecen un grupo de roles para asignar los diferentes permisos para su acceso. Los usuarios del sistema son aquellas personas que interactúan con la aplicación. Para estos existen algunas restricciones específicas. A continuación se describen los roles del sistema con sus responsabilidades.

Tabla 1 Roles del sistema

Roles	Responsabilidades
Administrador	Tendrá el control total del sistema, quien creará los usuarios según el nivel al que pertenezcan, ya sea municipal o provincial correspondiendo con los tipos de roles antes mencionados. También este será el responsable de realizar cualquier configuración necesaria para el correcto funcionamiento del sistema.
Funcionario municipal	Registra los boletines del día, posteriormente realizaría la codificación y registro de las opiniones, luego de finalizar estas operaciones exporta las opiniones correspondientes al boletín a enviar para la provincia.
Especialista provincial	Importan la información en la base de datos, para realizar una recodificación de las opiniones y crear a partir de estas las opiniones tipos, para posteriormente enviar el boletín hacia el CESPO.

Capítulo 2: Propuesta de solución

2.2 Listas de reservas del producto

Las características del sistema son propiedades o cualidades que el producto debe tener para satisfacer las necesidades del cliente. Complementan los aspectos fundamentales del mismo, describiendo un comportamiento funcional y visible por el usuario. Son propiedades del producto que lo hacen rápido, confiable, atractivo y usable (Beck, 1999).

2.2.1 Requisitos Funcionales

Para el desarrollo de un sistema de software es necesario realizar la captura de requisitos, con el objetivo de conocer las necesidades que los clientes plantean para resolver sus problemas, así como las propiedades o restricciones deseadas para que el equipo de trabajo pueda desarrollar un sistema que cumplan con las expectativas de los clientes.

A continuación se enumeran los requisitos funcionales:

R 1 Gestionar usuario

R 1.1 Autenticar usuario

R 1.2 Registrar usuario

R 1.3 Mostrar usuarios

R 1.4 Editar datos de los usuarios

R 1.5 Cambiar contraseña de los usuarios

R 1.6 Eliminar usuario

R 1.7 Cerrar sesión

R 2 Gestionar opinión

R 2.1 Registrar opinión

R 2.2 Mostrar opiniones

R 2.3 Editar datos de las opiniones

R 2.4 Eliminar opinión

R 3 Gestionar boletín

R 3.1 Registrar boletines

R 3.2 Mostrar boletines

Capítulo 2: Propuesta de solución

R 3.3 Editar datos de los boletines

R 3.4 Eliminar boletín

R 4 Gestionar activista

R 4.1 Registrar activista

R 4.2 Mostrar activistas

R 4.3 Editar datos de los activistas

R 4.4 Eliminar activista

R 5 Evaluar activista de acuerdo a la cantidad de opiniones y la sistematicidad con que las entregó

R 6 Mostrar evaluaciones de los activistas

R 7 Exportar boletín

R 8 Importar archivo.

R 9 Crear opinión tipo

R 10 Asignar opinión tipo

R 11 Eliminar opinión tipo

R 12 Mostrar opiniones tipos

R 13 Mostrar la cantidad de opiniones por tema, subtema, asunto, subasunto, red, tipo de boletín y activista

2.2.2 Aspectos no funcionales

Los requisitos no funcionales son las cualidades que el producto debe tener y que lo hace rápido, usable, fiable y seguro. A continuación se exponen los requisitos no funcionales definidos:

1. Software: Para instalar el sistema es necesario:

Servidor:

- Sistema Operativo: Windows o Linux.
- Servidor web: Nodejs.
- Lenguaje JavaScript 1.8.5 o superior, y HTML 5.
- Gestor de base de datos: SQLite versión 3.

Capítulo 2: Propuesta de solución

Cliente:

- Sistema operativo: Windows XP o superior, cualquier distribución de Linux.
- Navegador web: Firefox versión 4.0 o superior, Google Chrome 27.0 o superior, Internet Explorer 7.0 o superior.

2. Hardware

- El servidor donde será alojado el sistema debe tener como mínimo: Procesador: 1 GHz, Memoria RAM: 256 MB, Capacidad de Disco Duro: 8 GB.

3. Rendimiento

- El sistema operará con grandes volúmenes de datos, por tanto, se hacen necesarios tiempos de respuestas cortos menos de 30 segundos y que procese la información lo más rápido posible.
- El servidor debe soportar como mínimo 3 usuarios conectados simultáneamente.

4. Seguridad

- Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.

5. Portabilidad

- El sistema debe ser multiplataforma.

6. Capacidad

- Considerar características técnicas mínimas para la ejecución en clientes. Soportar múltiples conexiones al sistema de manera simultánea. Considerar el crecimiento esperado en el volumen de datos.

2.3 Modelo conceptual

Un modelo conceptual es una representación gráfica de los objetos del sistema. Puede incluir conceptos y datos, así como las relaciones importantes entre ellos. Es utilizado para expresar el entendimiento sobre el negocio.

El equipo de trabajo decidió generar este artefacto, porque ayuda a los desarrolladores, usuarios y personas interesadas a utilizar un mismo vocabulario para entender la propuesta

Capítulo 2: Propuesta de solución

de solución; una vez identificado los conceptos y objetos relacionados con la codificación y análisis de las opiniones

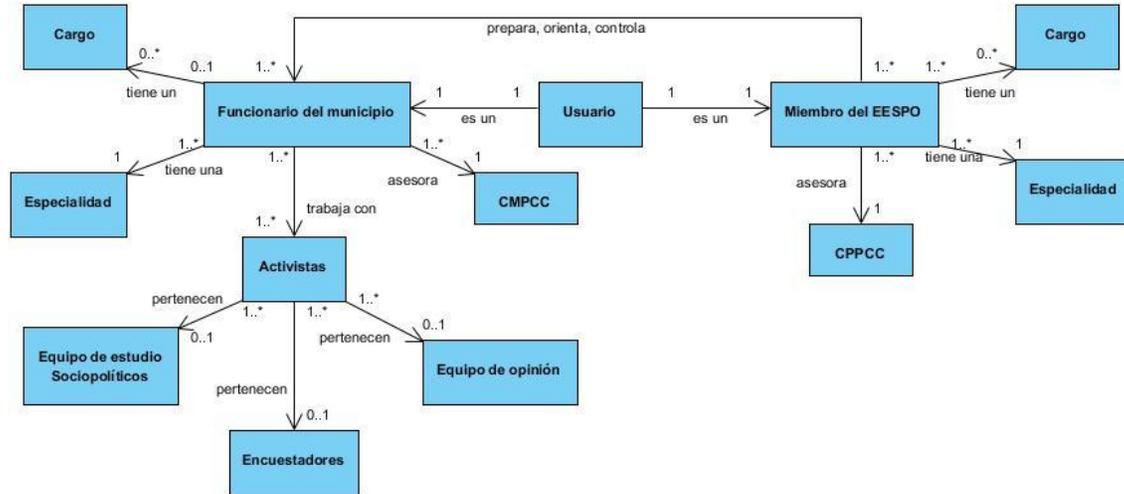


Figura 2 Diagrama conceptual del negocio

A continuación se muestran los principales conceptos que se manejan en el desarrollo del sistema:

- Usuario: es aquella persona que accede al sistema delimitado por los privilegios definidos para rol con el que fue creado.
- Funcionario del municipio: especialista que trabaja en el comité municipal del PCC encargado de la opinión del pueblo.
- CMPCC: comité municipal del PCC.
- Miembro del EESPO: especialista que trabaja en el comité provincial del PCC encargado de la opinión del pueblo.
- CPPCC: comité provincial del PCC.
- Activistas: son los encargados de la recepción de las opiniones espontáneas de la población y conforman una red en cada municipio.
- Equipo de estudios sociopolíticos: cumple la misión de asesorar, a través de las investigaciones o estudios sociopolíticos, a la dirección municipal del Partido, y participa en la preparación teórica y metodológica de los activistas y encuestadores.
- Encuestadores: encargados de la aplicación de diferentes técnicas como los sondeos, encuestas, entrevistas y guías de observación, a través de los cuales se recopila la información. En los estudios o investigaciones que realiza el Partido, ya

Capítulo 2: Propuesta de solución

sean de carácter nacional, provincial o local, sobre los encuestadores recae una gran responsabilidad, de la calidad de su trabajo depende la confiabilidad de los resultados finales de la investigación.

- Equipo de opinión: auxilia al funcionario en las tareas diarias del boletín de opiniones espontáneas, con el análisis y codificación de las opiniones, los partes diarios al Equipo Provincial y la elaboración de boletines periódicos de opiniones espontáneas que se circulan al Buró Municipal.
- Cargo: jerarquía de los miembros que trabajan en el PCC.
- Especialidad: área de trabajo sobre la que tienen dominio los especialistas del PCC.

2.4 Historia de Usuarios

Las Historias de Usuarios (HU) son tarjetas de papel en las cuales el cliente describe las características del sistema. En cualquier momento del desarrollo pueden ser modificadas o reemplazadas por otras más generales o específicas, o incluso añadir nuevas. Cada una es lo suficientemente entendible y está delimitada como para que los desarrolladores puedan implementarlas en unas semanas (Jeffries, 2001). Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia (Castillo Oswaldo, y otros, 2008).

Las HU son utilizadas para representar una breve descripción del comportamiento del sistema, sin usar un lenguaje técnico realizándose una por cada requisito funcional, empleadas para hacer estimaciones de tiempo y planes de entrega. A continuación se especifican los aspectos definidos por la Universidad de las Ciencias Informáticas (UCI) (Especificación de Requisitos de Software) que deben contener las HU:

- Número: número asignado a la HU.
- Nombre: nombre de la HU.
- Programador: persona que implementa la HU.
- Prioridad: nivel de prioridad de la HU.
- Riesgo en desarrollo: hace referencia a los riesgos identificados en plan de riesgos.
- Iteración Asignada: número de iteraciones.
- Tiempo estimado: estimación del tiempo de duración de la HU acordada por el equipo de desarrollo, una semana de trabajo equivale a 5 días hábiles, trabajando 40 horas, es decir, 8 horas diarias.

Capítulo 2: Propuesta de solución

- Tiempo real: tiempo real dedicado a la realización de la HU en semanas.
- Descripción: posee una breve descripción de lo que realizará la HU.
- Observaciones: aclaraciones necesarias.
- Prototipo de interfaz: son elementos de diseño visual que permiten al usuario tener una idea de las interfaces que mostrará el sistema.

A continuación se muestran las descripciones de las HU:

Tabla 2 HU Gestionar usuario

Historia de Usuario	
Número: 1	Nombre: Gestionar usuario
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Real: 1 semana
Descripción: El sistema muestra una vista a través de la cual el administrador puede registrar, mostrar, editar, cambiar contraseña y eliminar usuarios. Además permite que el funcionario municipal y especialista provincial puedan autenticarse y obtener los privilegios según su rol, cambiar sus contraseñas y cerrar su sesión. La interfaz registrar está compuesta por los campos nombre y apellidos, usuario, tipo de usuario, contraseña y repetir contraseña. La interfaz mostrar contiene varios campos entre los que se encuentran usuario, email, además de un buscador general cuya información se visualizará en una tabla, que posee las opciones editar y eliminar.	
Observaciones: El usuario deberá estar autenticado con el rol administrador, además el sistema le indicará los campos que son obligatorios para gestionar a los usuarios. Al crear un usuario se le asignará un rol el cual tendrá sus responsabilidades en el sistema.	

Capítulo 2: Propuesta de solución

Tabla 3 HU Gestionar opinión

Historia de Usuario	
Número: 2	Nombre: Gestionar opinión
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Real: 1 semana
Descripción: El sistema permitirá que el administrador y el funcionario municipal puedan registrar, mostrar, editar y eliminar opiniones. La interfaz registrar está compuesta por los campos opinión y descripción; también contiene los atributos para seleccionar el asunto, subasunto, sentido, red, boletín y activista. La interfaz de mostrar cuenta con el conjunto de filtros que permiten visualizar las opiniones que el usuario desee en la tabla que se encuentra debajo de los mismos; la cual permite editar y eliminar opiniones.	
Observaciones: El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal, además el sistema le indicará los campos que son obligatorios para gestionar las opiniones.	

Tabla 4 HU Gestionar boletín

Historia de Usuario	
Número: 3	Nombre: Gestionar boletín
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Real: 1 semana
Descripción: Una vez que el administrador o el funcionario municipal estén dentro del sistema pueden ejecutar las acciones registrar, mostrar, editar y eliminar boletín. La interfaz registrar presenta dos campos de texto en el que se registrará el nombre del boletín y el tipo de boletín, así como un botón para seleccionar la fecha. La interfaz mostrar contiene un buscador general que permite visualizar los boletines en una tabla, la cual posee en sus dos últimas columnas las opciones editar y eliminar boletines.	
Observaciones: El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal, además el sistema le indicará los campos que son obligatorios para gestionar un boletín.	

Capítulo 2: Propuesta de solución

Tabla 5 HU Gestionar activista

Historia de Usuario	
Número: 4	Nombre: Gestionar activista
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Real: 1 semana
Descripción: El sistema posibilita registrar, mostrar, editar y eliminar activistas. Registrar tiene un campo de texto en el que se introducirá el nombre y apellido, también contiene varios campos entre los que se pueden evidenciar el código, el número de teléfono del trabajo, la dirección, el centro de trabajo. La interfaz mostrar posee un buscador general mediante el cual se visualizará el resultado en una tabla, en la que se podrá editar o eliminar los activistas necesarios.	
Observaciones: El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal, además el sistema le indicará los campos que son obligatorios para gestionar un activista.	

Tabla 6 HU Exportar boletín

Historia de Usuario	
Número: 7	Nombre: Exportar boletín
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Real: 1 semanas
Descripción: El sistema muestra una vista compuesta por un campo que permite seleccionar el boletín que se desea exportar, además de un buscador general cuya información se visualizará en una tabla.	
Observación: El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal, el exportar boletín se realizará según un intervalo de fecha exportando las opiniones correspondientes.	

Tabla 7 HU Importar boletín

Historia de Usuario	
Número: 8	Nombre: Importar boletín
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Reales: 1 semana
Descripción: El sistema permite que el administrador y el especialista provincial puedan importar el boletín enviado por el municipio, donde una vez seleccionada la opción podrán buscar el boletín deseado.	
Observaciones: El usuario deberá estar autenticado con el rol administrador o el de funcionario provincial, el importar boletín se realizará cargando el boletín enviado por el funcionario del municipio.	

Capítulo 2: Propuesta de solución

Tabla 8 HU Crear opinión tipo

Historia de Usuario	
Número: 9	Nombre: Crear opinión tipo
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 3
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Real: 1 semanas
Descripción: El sistema permitirá que el administrador y el especialista provincial puedan crear a partir de las opiniones registradas en el sistema, las opiniones tipo.	
Observaciones: El usuario deberá estar autenticado con el rol administrador o el de especialista provincial.	

Se definieron un total de 13 HU, en el epígrafe se plasman las principales del proceso del negocio, las demás se pueden encontrar en el Anexo 1.

2.5 Estimación de esfuerzo por HU

La estimación de esfuerzos las establecen los programadores utilizando las semanas como medida. Las HU deben ser programadas en un tiempo estimado hasta 3 semanas. Si la estimación supera las 3 semanas la HU deberá ser dividida hasta que pueda ser desarrollada en un tiempo factible. En caso de que el esfuerzo sea menor de 1 semana la HU será combinada con otra historia (Joskowicz, 2008).

Tabla 9 Estimación de esfuerzo por HU

Número	Historias de usuarios	Puntos de estimación
1	Gestionar usuario	1
2	Gestionar opinión	1
3	Gestionar boletín	1
4	Gestionar activista	1
5	Evaluar activista	1
6	Mostrar evaluaciones	0.6
7	Exportar boletín	1
8	Importar archivo	1
9	Crear opinión tipo	1
10	Asignar opinión tipo	1
11	Eliminar opinión tipo	0.4
12	Mostrar opinión tipo	0.6
13	Mostrar cantidad de opiniones	1

Capítulo 2: Propuesta de solución

2.6 Planificación del proyecto

XP plantea la planificación como un constante diálogo entre el cliente y los desarrolladores donde los primeros definen el alcance del proyecto, la prioridad de las HU, la composición de las versiones y la fecha de entrega de las mismas, y los desarrolladores estiman el tiempo necesario para implementar cada HU, el cual debe ser de entre una y tres semanas.(Joskowicz, 2008).

2.5.1 Iteraciones

Los proyectos que utilizan como guía para su desarrollo a XP, se deben planificar en iteraciones de aproximadamente tres semanas de duración. Para cada iteración se define un conjunto de HU que se van a implementar. En la primera iteración se seleccionan las HU que abarquen los aspectos más importantes de la arquitectura global, es decir, las HU que hagan referencia a la construcción de la estructura de todo el sistema. (Calabria, 2003).

La siguiente investigación se divide en iteraciones, de manera que se realicen entregas parciales. En el plan de iteraciones se define el orden de implementación de las HU en cada iteración conjuntamente con el tiempo de trabajo de las mismas.

Tabla 10 Plan de iteraciones

Iteración	Orden de las HU a implementar.	Tiempo de trabajo
1ra iteración	HU 1 Gestionar usuario HU 2 Gestionar opinión HU 3 Gestionar boletín	3 semanas
2da iteración	HU 4 Gestionar activista HU 5 Evaluar activistas HU 6 Mostrar evaluaciones de los activistas	3 semanas
3ra iteración	HU 7 Exportar boletín HU 8 Importar archivo HU 9 Crear opinión tipo	3 semanas
4ta iteración	HU 10 Asignar opinión tipo HU 11 Eliminar opinión tipo HU 12 Mostrar opinión tipo HU 13 Mostrar la cantidad de opiniones por tema, subtema, asunto, subasunto, red, tipo de boletín y activista	3 semanas

Capítulo 2: Propuesta de solución

2.5.2 Plan de entregas

El plan de entregas es un documento importante, ya que establece un compromiso entre el equipo de desarrollo y los clientes sobre la entrega real de la solución, en el que se recogen las HU para conformar una entrega y el orden de las mismas. Luego de una reunión con el cliente, considerando las necesidades planteadas y las estimaciones reales por los programadores se determinó el siguiente Plan de Entregas:

Tabla 11 Plan de entregas

Entregable	1ra entrega (2da semana de Febrero)	2da entrega (2da semana de Marzo)	3ra entrega (2da semana de Abril)	4ta entrega (2da semana de Mayo)
Sistema de gestión de información para el departamento de opinión pública del PCC de Cienfuegos	Versión 0.1	Versión 0.2	Versión 0.3	Versión 0.4

Tabla 11 Funcionalidades disponibles por entrega del producto

Historias de Usuario	1ra entrega	2da entrega	3ra entrega	4ta entrega
HU 1 Gestionar usuario	X			
HU 2 Gestionar opinión	X			
HU 3 Gestionar boletín	X			
HU 4 Gestionar activista		X		
HU 5 Evaluar activistas		X		
HU 6 Mostrar evaluaciones de los activistas		X		
HU 7 Exportar boletín			X	
HU 8 Importar archivo			X	
HU 9 Crear opinión tipo			X	
HU 10 Asignar opinión tipo				X
HU 11 Eliminar opinión tipo				X
HU 12 Mostrar opinión tipo				X
HU 13 Mostrar la cantidad de opiniones por tema, subtema, asunto, subasunto, red, tipo de boletín y activista				X

Capítulo 2: Propuesta de solución

2.6 Diseño de la aplicación

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar (Castillo Oswaldo, y otros, 2008).

En esta fase se definen los artefactos necesarios para guiar el proceso de desarrollo, así como los modelos de datos. La herramienta a utilizar para obtener la aplicación será Node.js, por solicitud y acuerdo con el cliente.

2.6.1 Tarjetas Clase-Responsabilidades-Colaboradores

Las tarjetas CRC utilizan la técnica de modelado basada en objetos, representando cada tarjeta CRC a un objeto, identificando las clases y sus responsabilidades. Las tarjetas están compuestas por el nombre de la clase colocado como título, en la parte izquierda se colocan las responsabilidades (funcionalidades) y en la parte derecha las clases que se implican en cada funcionalidad (Joskowicz, 2008). A continuación se muestran algunas de las tarjetas CRC para la propuesta de solución:

Tabla 12 Tarjeta CRC Gestionar usuario

Clase: Usuario	
Responsabilidad	Colaboración
user_controller.insertUser(userData)	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.updateUser(userData)	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.loguinUser(user, passwr)	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.actualiza_Passwr(user, pass)	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getUserById(id_user)	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.listadoUsuarios(rol)	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.eliminarUsuario(user)	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getUsers()	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.userAccess(roless)	Crypto, Moment, SQLite3, Express, HTTP, Path

Capítulo 2: Propuesta de solución

user_controller.updateVisitas(v)	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getVisitas()	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getTotalUsuarios()	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getTotalOpiniones()	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getTotalAct()	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getRoles()	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getProvinciaUsers()	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getMunUsers(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
user_controller.getDatosUsers(id)	Crypto, Moment, SQLite3, Express, HTTP, Path

Capítulo 2: Propuesta de solución

Tabla 13 Tarjeta CRC Gestionar opinión

Clase: Opinión	
Responsabilidad	Colaboración
opinion_controller.getTema()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getSubTema()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getOpiniones()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getSubAsuntos()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getSentido()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getRed()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getBoletin()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getActivista()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getProvincia()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.listadoOpiniones(filtrado, fil_fecha,rol)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getProOpinion(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.insertarOpinion(datos)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getTemaSubtema(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getTemaSubAsunto(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getTema()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getSubtemaAsunto(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.updateOpinion(op)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.obtenerFecha(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.obtenerSubtema(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.obtenerTema(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getMunOpinion(id)	Crypto, Moment, SQLite3, Express, HTTP, Path

Capítulo 2: Propuesta de solución

opinion_controller.getDatosOpinion(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getBoletin()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getAsuntos()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.getActivista()	Crypto, Moment, SQLite3, Express, HTTP, Path
opinion_controller.eliminarOpinion(op)	Crypto, Moment, SQLite3, Express, HTTP, Path

Tabla 14 Tarjeta CRC Gestionar boletín

Clase: Boletín	
Responsabilidad	Colaboración
boletin_controller.insertarBoletin(datos)	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.updateBol(bol)	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.getBoletin()	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.listadoBoletines()	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.eliminarBoletin(bol)	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.getBoletin2()	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.getDatosBoletin(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.getExportarBoletin(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.getOp(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.getOp2()	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.getTipoBoletin()	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.listadoTablas(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.listadoExOp(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
boletin_controller.listadoExOp2()	Crypto, Moment, SQLite3, Express, HTTP, Path

Capítulo 2: Propuesta de solución

Tabla 15 Tarjeta CRC Gestionar activista

Clase: Activista	
Responsabilidad	Colaboración
activista_controller.getMunicipiosProv(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getCPMunicipios(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.insertarActivista(datos)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getProvincia()	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getMunicipio2()	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getCP()	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getActivista()	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getActivista2()	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.listadoActivistas()	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.eliminarActivista(act)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.updateActivista(act)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getCantOpiniones(filtro1)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getCpAct(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getMunAct(id)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getSectores()	Crypto, Moment, SQLite3, Express, HTTP, Path

Tabla 16 Tarjeta CRC Exportar boletín

Clase: Boletín	
Responsabilidad	Colaboración
Boletin_controller.getExportarTablas	Crypto, Moment, SQLite3, Express, HTTP, Path

Tabla 17 Tarjeta CRC Importar archivo

Clase: Boletín	
Responsabilidad	Colaboración
provincia_controller.leerArchivo(datos)	Crypto, Moment, SQLite3, Express, HTTP, Path

Capítulo 2: Propuesta de solución

En este epígrafe se muestran las tarjetas CRC: Gestionar usuario, Gestionar opinión, Gestionar boletín, Gestionar activista, Exportar boletín, Importar archivo, correspondientes a las HU analizadas anteriormente, las restantes se encuentran en el Anexo 2.

2.6.2 Propuesta de arquitectura

La propuesta de arquitectura del presente trabajo está basado en el patrón arquitectónico Modelo – Vista – Controlador (MVC), el cual separa la lógica en tres capas (Potencier, 2009):

La capa del Modelo define la lógica de negocio, es la responsable de acceder a la capa de almacenamiento de datos. Las clases y archivos relacionados con el modelo se guardan en el directorio /pcc_nodejs_opiniones/models. La clase install se encarga de crear todas las tablas en la base de datos e insertar los valores por defecto como: tema, subtema, asunto, subasunto, sentido, entre otros.

La Vista es con lo que el usuario interactúa. Es responsable de recibir los datos del modelo y la controladora y mostrarlos al usuario. Estas son guardadas en la carpeta /pcc_nodejs_opiniones/views, donde la vista principal es admin.js la cual carga todos los enlaces y scripts de las páginas.

El Controlador invoca al Modelo para obtener algunos datos que le envía a la Vista para la presentación al cliente. En la aplicación todas las solicitudes son gestionadas por varios controladores entre los que se encuentran user-controller, opinion-controller, entre otros, siendo el principal app.js, encargado de crear el servidor y el puerto de escucha, la dirección de las carpetas donde estarán las vistas y los CSS, además del motor de plantilla a utilizar.

En la figura 3 se puede evidenciar la representación del patrón arquitectónico MVC de la aplicación.

Capítulo 2: Propuesta de solución

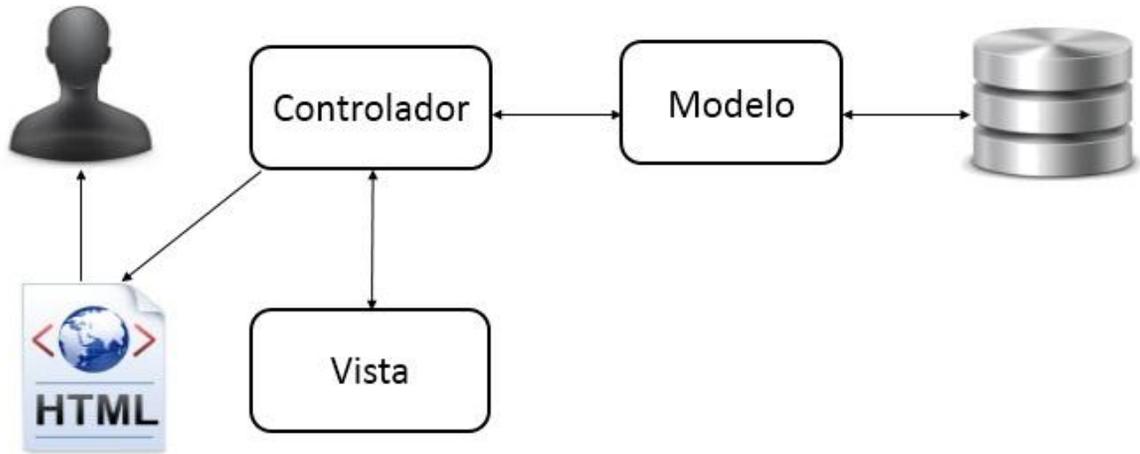


Figura 3 Representación del patrón arquitectónico MVC

2.6.3 Modelo de datos

En este subepígrafe se muestra el modelo de datos de la propuesta de solución, el mismo fue desarrollado en la herramienta CASE Visual Paradigm for UML, en el que se aprecian las relaciones entre las capas.

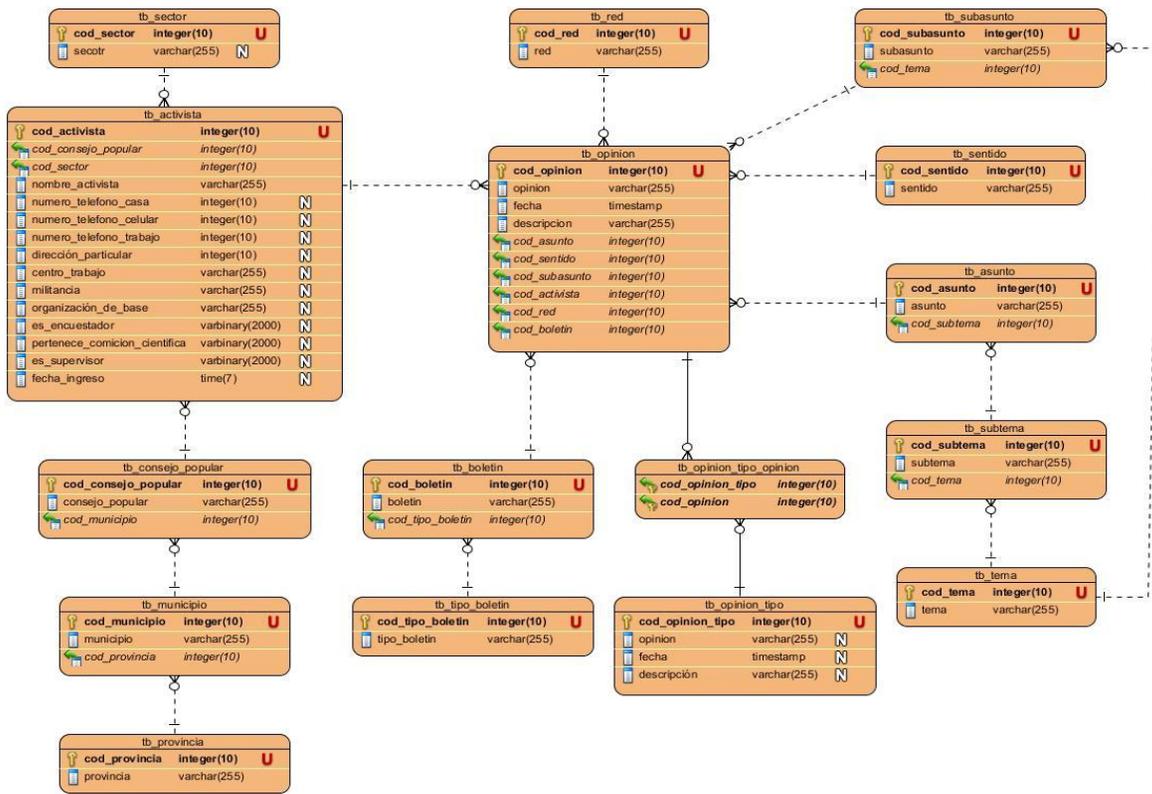


Figura 4 Modelo físico de la base de datos de la propuesta de solución

Capítulo 2: Propuesta de solución

Seguidamente se muestra una descripción de la información almacenada en las tablas de la base de datos que utiliza la aplicación:

- tb_red: redes de activistas que existen en el PCC.
- tb_sentido: sentidos en que se clasifican las opiniones.
- tb_sector: sectores a los que los activistas pertenecen.
- tb_consejo_popular: consejos populares del país.
- tb_municipio: municipios del país.
- tb_provincia: provincias del país.
- tb_activistas: activistas que recogen las opiniones.
- tb_tema: temas en los que se clasifican las opiniones.
- tb_subtema: subtemas en los que se clasifican las opiniones.
- tb_asunto: asuntos en los que se clasifican las opiniones.
- tb_subasunto: subasuntos en los que se clasifican las opiniones.
- tb_tipo_boletin: tipos de boletines en que se pueden clasificar los boletines.
- tb_boletin: boletines a los que van a pertenecer las opiniones.
- tb_opinion: opiniones recogidas por los activistas, así como la codificación de las mismas.

2.6.4 Patrones de diseño

Craig Larman define en la segunda edición de su libro “UML y Patrones” a un patrón como “un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos” (Larman, 2004). Los patrones de diseño comunican los estilos y soluciones consideradas como "buenas prácticas", que los expertos en el diseño orientado a objetos utilizan para la creación de sistemas (Larman, 2004).

Los patrones de diseños se dividen en 2 grandes grupos los GRASP (Responsibility Assignment Software Patterns) (Patrones Generales de Software para Asignar Responsabilidades) y los GOF (Gang of Four).

2.6.4.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Larman, 2004). En la solución se utilizaron los siguientes patrones.

Capítulo 2: Propuesta de solución

Experto: Este patrón resuelve el problema de “asignar una responsabilidad al experto en información -la clase que tiene la información necesaria para realizar la responsabilidad” (Larman, 2004). Es utilizado en varias clases un ejemplo de ellas es user- controller.js. Clase que contiene toda la información necesaria de los usuarios.

Alta cohesión: Este patrón resuelve el problema de “asignar una responsabilidad de manera que la cohesión permanezca alta” (Larman, 2004). La alta cohesión se evidencia en varias clases como activista – controller.js, donde la información que se almacena en esta clase está relacionada con la misma.

Controlador: Este patrón resuelve el problema de asignar la responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase. Clase que sirve de intermediaria entre una interfaz y la acción que se desee ejecutar (Larman, 2004). De tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Dentro del sistema está presente en el controlador principal app.js.

2.6.4.2 Patrones GOF

Singleton: Es un patrón de tipo creación, ya que abstrae el proceso de creación de instancias. Resuelve el problema de que exista una instancia única de una clase, proporcionando un punto de acceso global a la misma (Larman, 2004). En el sistema se evidencia en la clase index.js del modelo, encargada de crear la base de datos a la cual pueden acceder las demás clases.

2.7 Interfaz de usuario

Una interfaz de usuario es la parte con la que el usuario puede comunicarse con el sistema. Suelen ser fáciles de entender y fáciles de interactuar, incluyen varios elementos como: pantallas, ventanas, menú, botones, documentación entre otros elementos, que permiten al usuario realizar determinada tarea.

A continuación se muestran las interfaces de usuario de las HU más significativas, las restantes se encuentran en el Anexo 3:

Capítulo 2: Propuesta de solución

Registrar Usuario
▼

Nombre y apellidos: *

usuario: *

Tipo de usuario: * Seleccione... ▼

Provincia: * Seleccione... ▼

Municipio: * Seleccione... ▼

Contraseña: *

Repetir contraseña: *

Registrar ↻

Figura 5 Interfaz de usuario Registrar usuario

Opciones de filtrado
▼

Rango de fecha a

Tema Seleccione... ▼

Subtema Seleccione... ▼

Provincia Seleccione... ▼

Municipio Seleccione... ▼

Consejo popular Seleccione... ▼

Asunto Seleccione... ▼

Subasunto Seleccione... ▼

Red Seleccione... ▼

Sentido Seleccione... ▼

Boletín Seleccione... ▼

Activista Seleccione... ▼

Datos de las opiniones

▼

Opinión	Tema	Subtema	Asunto	Subasunto	Sentido	Fecha	Descripción	Activista	Red	Boletín	Editar	Eliminar
Opinión	Tema	Subtema	Asunto	Subasunto	Sentido	Fecha	Descripción	Activista	Red	Boletín	Editar	Eliminar

Figura 6 Interfaz de usuario Mostrar opinión

Datos de los boletines
▼

10 ▼ Num Buscar

Nombre boletín	Tipo boletín	Fecha	Editar	Eliminar
Boletín1	Vida Cotidiana	04-20-2015	Editar	Eliminar
Nombre boletín	Tipo boletín	Fecha	Editar	Eliminar

1 de 1 en 1 Resultados < 1 >

Figura 7 Interfaz de usuario Mostrar boletín

Capítulo 2: Propuesta de solución

Opiniones	Fecha	Descripción	Tema	Sub_tema	Asunto	Sub_asunto	Sentido	Activista	Red	Boletín
Opiniones	Fecha	Descripción	Tema	Sub_tema	Asunto	Sub_asunto	Sentido	Activista	Red	Boletín

Figura 8 Interfaz de usuario Exportar boletín

Examinar... No se ha seleccionado ningún archivo.
Seleccione el archivo.

Figura 9 Interfaz de usuario Importar archivo

Conclusiones del capítulo

La selección de XP como metodología de desarrollo del Sistema de Gestión de Información para el departamento de opinión pública del PCC de Cienfuegos permitió desarrollar el análisis y diseño del sistema.

- En el estudio realizado se determinó que la organización contará con 3 tipos de usuarios para la interacción con el sistema, ya que serán los responsables de llevar a cabo los procesos de codificación y análisis de las opiniones.
- Se definieron 13 HU distribuidas en 4 iteraciones, que describen los aspectos principales a tener en cuenta para el desarrollo del sistema.
- El patrón arquitectónico utilizado es el MVC, con el objetivo de separar los datos y la lógica del negocio.
- Se construyó el plan de entregas, especificándose el orden de entregas que se realizaron, además de 13 tarjetas CRC.

Capítulo 3: Implementación y Prueba

Capítulo 3: Implementación y prueba

Uno de los pilares de XP es el uso de pruebas para comprobar las funcionalidades implementadas, por lo que XP propone probar constantemente el código. Se realiza un desarrollo dirigido por pruebas, donde el código debe vencer las pruebas establecidas con anterioridad para considerarse válido. Esto permite verificar el funcionamiento del sistema antes de ser entregado a los usuarios finales.

En el capítulo se abordarán los elementos de la implementación y prueba de la propuesta de solución, en el que se muestran las tareas de ingeniería, la estrategia de pruebas del producto, especificándose las pruebas realizadas y los resultados arrojados.

3.1 Implementación

La fase de implementación, una de las fundamentales en el desarrollo de XP es guiada por la programación en pareja, ya que permite obtener un código eficiente y con calidad. En la misma se desarrollan las funcionalidades del sistema y se comprueba su correcto funcionamiento antes de ser entregado a los usuarios finales. Durante esta etapa el equipo de trabajo debe mantener constante comunicación con el cliente, por lo que se recomienda un representante del cliente, donde la empresa colaboró con un especialista del departamento de opinión pública.

3.1.1 Estándares de codificación

Los estándares de codificación son un conjunto de reglas a seguir por los desarrolladores con el objetivo de establecer un orden y un formato común en el código fuente del software en desarrollo. Son muy importantes para la etapa de construcción del software ya que permite que el personal del proyecto pueda entender de forma fácil el código, garantizándose la organización y estructura del código fuente (Microsoft, 2014).

Para el desarrollo de la solución se utilizaron los estándares de codificación CamelCase y Documentación con el objetivo de ajustar las nomenclaturas en la implementación del sistema y obtener un producto organizado.

CamelCase

Es un estándar en el que los nombres de los identificadores están compuestos por una o más palabras juntas, iniciando cada palabra con letra mayúscula y el resto en minúscula,

Capítulo 3: Implementación y Prueba

donde la primera letra del identificador no es mayúscula sino minúscula. Este estándar fue seleccionado para el nombramiento de las funcionalidades, la figura 10 ejemplifica el uso del CamelCase:

```
boletin_controller.insertarBoletin = function(datos, callback) {  
    stmt = db.prepare("SELECT * FROM boletin WHERE name_boletin = ?");  
    stmt.bind(datos.nb);  
    stmt.get(function(error, row) {  
        if (error) {  
            callback(error);  
        }  
    });  
};
```

Figura 10 Estándar de codificación CamelCase

Documentación

Los comentarios en el código de cualquier aplicación son utilizados para aclarar determinados fragmentos de código, suelen escribirse al principio de cada función. Se deben escribir frases completas, comenzándolas con mayúscula. Se utilizan las etiquetas `/* */` para comentarios en varias líneas y `//` para comentarios de una única línea.

La figura 11 ejemplifica el uso de comentarios de una única línea.

```
//Este método elimina un usuario de la BD a través de la sentencia DELETE  
user_controller.eliminarUsuario = function(user, callback) {  
    db.run("DELETE FROM users WHERE id = $id", {  
        $id: user.id  
    }, callback('ok', 200));  
};
```

Figura 11 Comentarios de una única línea

La figura 12 ejemplifica el uso de comentarios en varias líneas.

Capítulo 3: Implementación y Prueba

```
/*Este método devuelve en un arreglo de objetos que contiene
 todos los usuarios en la BD con la función db.all*/
user_controller.getUsers = function(callback) {
  db.all("SELECT * FROM users", function(err, rows) {
    if (err) {
      throw err;
    }
    else {
      callback(null, rows);
    }
  });
};
```

Figura 12 Comentarios en varias líneas

3.1.2 Tareas de Ingeniería

Las HU fueron desglosadas en tareas de ingeniería (TI), las cuales les facilita la implementación a los programadores. Las tareas tienen una duración de entre uno y tres días y podrán ser verificadas a través de las pruebas. A continuación se muestran las TI elaboradas por cada HU y los elementos que se tuvieron en cuenta para la realización de cada una de ellas:

- **Número tarea:** número de la tarea que se esté realizando.
- **Número historia:** identifica la HU.
- **Nombre tarea:** identifica la TI.
- **Tipo de tarea:** desarrollo/ corrección/ mejora/ otra.
- **Estimación:** duración de la implementación en días.
- **Fecha inicio:** comienzo de la tarea.
- **Fecha fin:** fin de la tarea.
- **Programador responsable:** persona encargada de la realización de la TI.
- **Descripción:** breve descripción de la TI

Capítulo 3: Implementación y Prueba

Tabla 18 TI Registrar usuario

Tarea	
Número: 1	Número de HU: 1
Nombre: Registrar usuario	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 11 de Enero 2015	Fecha fin: 12 de Enero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad registrar usuario de la clase usuario.	

Tabla 19 TI Mostrar usuarios

Tarea	
Número: 4	Número de HU: 1
Nombre: Mostrar usuarios	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 20 de Enero 2015	Fecha fin: 21 de Enero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad mostrar usuarios, de la clase usuario.	

Tabla 20 TI Registrar opinión

Tarea	
Número: 8	Número de HU: 2
Nombre: Registrar opinión	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 29 de Enero 2015	Fecha fin: 30 de Enero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad insertar opinión, de la clase opinión.	

Tabla 21 TI Mostrar opinión

Tarea	
Número: 11	Número de HU: 2
Nombre: Mostrar opinión	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 9 de Febrero 2015	Fecha fin: 10 de Febrero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: implementar la funcionalidad mostrar opinión, de la clase opinión.	

Capítulo 3: Implementación y Prueba

Tabla 22 TI Registrar boletín

Tarea	
Número: 12	Numero de HU: 3
Nombre: Registrar boletín	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 12 de Febrero 2015	Fecha fin: 13 de Febrero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad insertar boletín, de la clase boletín.	

Tabla 23 TI Mostrar boletines

Tarea	
Número: 13	Numero de HU: 3
Nombre: Mostrar boletines	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 15 de Febrero 2015	Fecha fin: 16 de Febrero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad mostrar boletines, de la clase boletín.	

Tabla 24 TI Registrar activista

Tarea	
Número: 16	Numero de HU: 4
Nombre: Registrar activista	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 26 de Febrero 2015	Fecha fin: 27 de Febrero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad registrar activista en el sistema, de la clase activista.	

Tabla 25 TI Mostrar activista

Tarea	
Número: 17	Numero de HU: 4
Nombre: Mostrar activistas	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 2 de Marzo 2015	Fecha fin: 3 de Marzo 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad mostrar activista, de la clase activista.	

Capítulo 3: Implementación y Prueba

Tabla 26 TI Exportar boletín

Tarea	
Número: 22	Numero de HU: 7
Nombre: Exportar boletín	
Tipo de Tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 20 de Marzo 2015	Fecha fin: 22 de Marzo 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad exportar boletín, de la clase boletín.	

Tabla 27 TI Importar archivo

Tarea	
Número: 23	Numero de HU: 8
Nombre: Importar archivo	
Tipo de Tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 28 de Marzo 2015	Fecha fin: 30 de Marzo 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad importar archivo de la clase boletín.	

Se desarrollaron un total de 30 tareas de ingeniería, en el epígrafe se muestran las tareas: Registrar usuario, Mostrar usuario, Registrar opinión, Mostrar opinión, Registrar boletín, Mostrar boletín, Registrar activista, Mostrar activista, Exportar boletín, Importar archivo, relacionadas con las HU que se describen en el Capítulo 2 de la investigación, las restantes se encuentran en el Anexo 4.

3.1.3 Diagrama de despliegue

El diagrama de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución. El objetivo del modelo de despliegue es mostrar la configuración de los elementos de hardware y software y las conexiones entre estos elementos en el sistema. El modelo contiene un servidor web donde estará instalado el sistema, este se conecta mediante los protocolos TCP/IP al servidor SQL3 (Servidor de la BD) donde estará guardada toda la información de la codificación y análisis de las opiniones. Las estaciones de trabajo (PC Cliente) se conectan al servidor web a través de los protocolos TCP/IP, donde los usuarios interactúan con el sistema.

En la figura 13 se muestra el diagrama de despliegue del sistema:

Capítulo 3: Implementación y Prueba

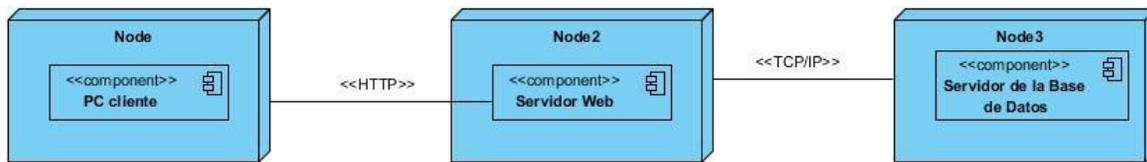


Figura 13 Diagrama de despliegue

3.2 Análisis del funcionamiento del sistema

Se realiza un análisis del funcionamiento del sistema para analizar aspectos que no han sido analizados anteriormente, que responden a los requisitos de seguridad, usabilidad y fiabilidad del sistema.

3.2.1 Validación de la entrada de datos

El sistema maneja información sensible por lo que es necesaria la validación de los datos que posteriormente se guardarán en la base de datos. Con el propósito de verificar que los datos registrados por los usuarios son correctos o no, y prevenir cualquier fallo del producto, para proporcionar seguridad e integridad de la información.

En la aplicación se realizan varias validaciones, a continuación se describen cada una de ellas:

- Validación de campos vacíos

Permite validar si faltan llenar algún campo, mostrando un mensaje con la definición: Este campo es obligatorio.

- Validación de números enteros

Se verifica que solamente se entren números enteros y no letras ni símbolos extraños.

- Validación de cadenas

Se verifica que solamente se entren letras y no otros símbolos extraños.

- Longitud de caracteres

Se valida la longitud de los caracteres para establecer el número mínimo de caracteres que se puede escribir en la celda.

Capítulo 3: Implementación y Prueba

3.2.2 Mensajes del sistema

Para el desarrollo de cualquier aplicación es importante tener en cuenta los tipos de mensajes que existen, con el objetivo de orientar a los usuarios de las acciones que están realizando y corregirlas en caso necesario.

El sistema cuenta con los mensajes de información, error y confirmación al usuario, a continuación se describen cada uno de ellos:

➤ Mensajes de información

Se muestran cuando los clientes realizan alguna acción que impliquen inserción, modificación o eliminación de un elemento, ejemplos de ellos son:

- Opinión insertada exitosamente.
- El usuario ya existe.
- Usuario modificado exitosamente.

En la figura 13 se muestra un mensaje de información del requisito Cambiar contraseña

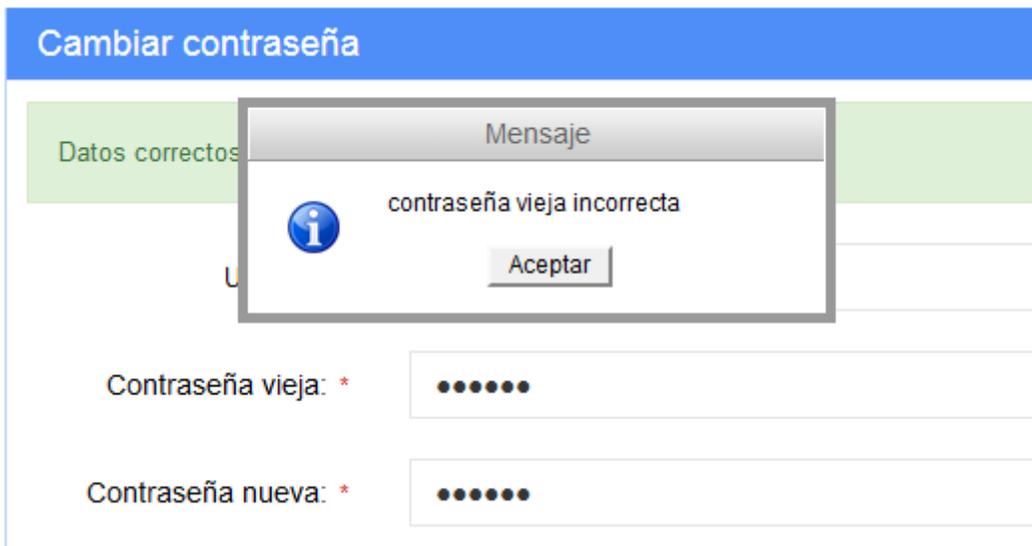


Figura 14 Mensaje de información Cambiar Contraseña

➤ Mensajes de error:

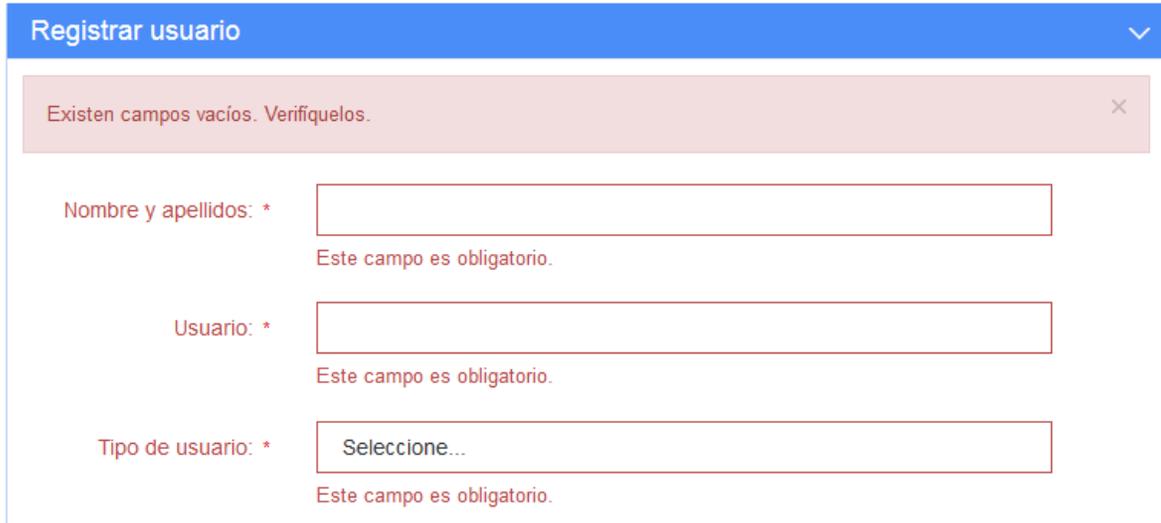
Indican un evento crítico que impide que se ejecute una acción. Los errores se generan cuando uno o más parámetros tienen valores no válidos. Dentro de estos mensajes se muestran:

- Este campo es requerido.

Capítulo 3: Implementación y Prueba

- Existen campos vacíos. Verifíquelos.

En la figura 14 se muestra un ejemplo del mensaje de error del requisito Registrar usuario.



The image shows a web form titled "Registrar usuario" with a blue header. Below the header is a red error message box that says "Existen campos vacíos. Verifíquelos." with a close button (X). The form contains three input fields, each with a red border and a red error message below it: "Nombre y apellidos: *" with "Este campo es obligatorio.", "Usuario: *" with "Este campo es obligatorio.", and "Tipo de usuario: *" with "Este campo es obligatorio." and a dropdown menu showing "Seleccione...".

Figura 15 Mensaje de error Registrar usuario

- Mensajes de confirmación:

Los mensajes de confirmación se muestran cuando es necesario asegurarse que el usuario desea realizar una acción deseada, por ejemplo: cuando se va a eliminar un usuario, es necesario asegurarse que eso es lo que desea, se verifica a través del mensaje:

- ¿Está seguro de eliminar este usuario?

Capítulo 3: Implementación y Prueba

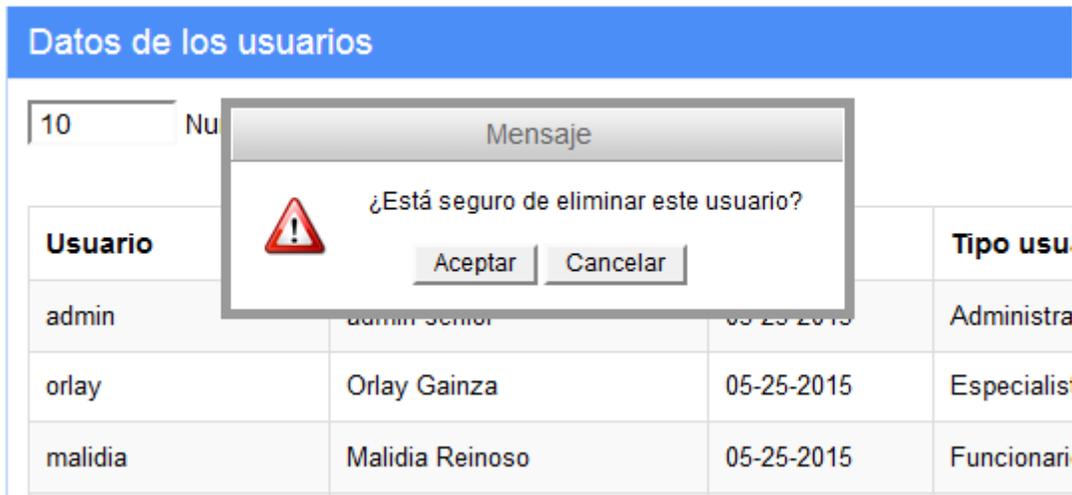


Figura 16 Mensaje de confirmación Eliminar boletín

3.3 Estrategias de pruebas aplicadas al sistema

Una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. Además proporciona una guía que describe los pasos a llevar a cabo como parte de la prueba (Ivar Jacobson, 2000).

3.3.1 Tipos de prueba

Existen dos tipos de pruebas que se realizan para asegurar que no existan errores en un software, estas son: las pruebas de Caja Negra y las pruebas de Caja Blanca. A continuación se describen cada una de ellas.

Pruebas de Caja Negra

Las pruebas de caja negra también conocidas como pruebas de comportamiento, se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente se ve como una "Caja Negra" cuyo comportamiento sólo puede ser determinado estudiando sus entradas y las salidas obtenidas a partir de ellas. Para realizarlas se selecciona un conjunto de posibles entradas y salidas sobre las que se realizarán las pruebas. Hay que tener en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo en el sistema, y como consecuencia producen una serie de salidas que revelan la presencia de defectos (Evaluación de Software, 2009).

Capítulo 3: Implementación y Prueba

Las pruebas de caja negra se aplicaron a las HU definidas anteriormente, para comprobar que las funciones están correctas y obtener la aceptación del cliente.

Pruebas de Caja Blanca

Las pruebas de caja blanca, denominada en ocasiones prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; ejecuten todos los ciclos en sus límites y con sus límites operacionales, y ejerciten las estructuras internas de datos para asegurar su validez (Wiley, 2001).

Con este método se determina cuáles son los casos de prueba a partir del código fuente del software y se utilizan las especificaciones para determinar el resultado esperado del caso. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa. La prueba de caja blanca del software se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o ciclos. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado (Wiley, 2001).

Las pruebas de caja blanca se aplicaron a las funcionalidades desarrolladas en el sistema.

3.3.2 Niveles de prueba

La estrategia que se ha de seguir a la hora de evaluar dinámicamente un sistema software debe permitir comenzar por los componentes más simples y más pequeños e ir avanzando progresivamente hasta probar todo el software en su conjunto. Más concretamente, los pasos a seguir son:

- Pruebas Unitarias: comienzan con la prueba de cada módulo.
- Pruebas de Integración: a partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces.

Capítulo 3: Implementación y Prueba

- Prueba del Sistema: el software ensamblado totalmente con cualquier componente hardware que requiere se prueba para comprobar que se cumplen los requisitos funcionales.
- Pruebas de Aceptación: el cliente comprueba que el software funciona según sus expectativas.

El sistema fue sometido a varias pruebas comprendidas en los niveles de pruebas de unidad y de aceptación. A continuación se describen las pruebas realizadas:

Pruebas unitarias

Las pruebas unitarias o pruebas de unidad deben ser construidas antes que el código, permitiéndole a los programadores tener máxima claridad de lo que van a programar antes de hacerlo, así como conocer cada uno de los casos de prueba que deberán pasar, lo que optimizará el trabajo y el código será de mayor calidad (Extreme Programming, 2009).

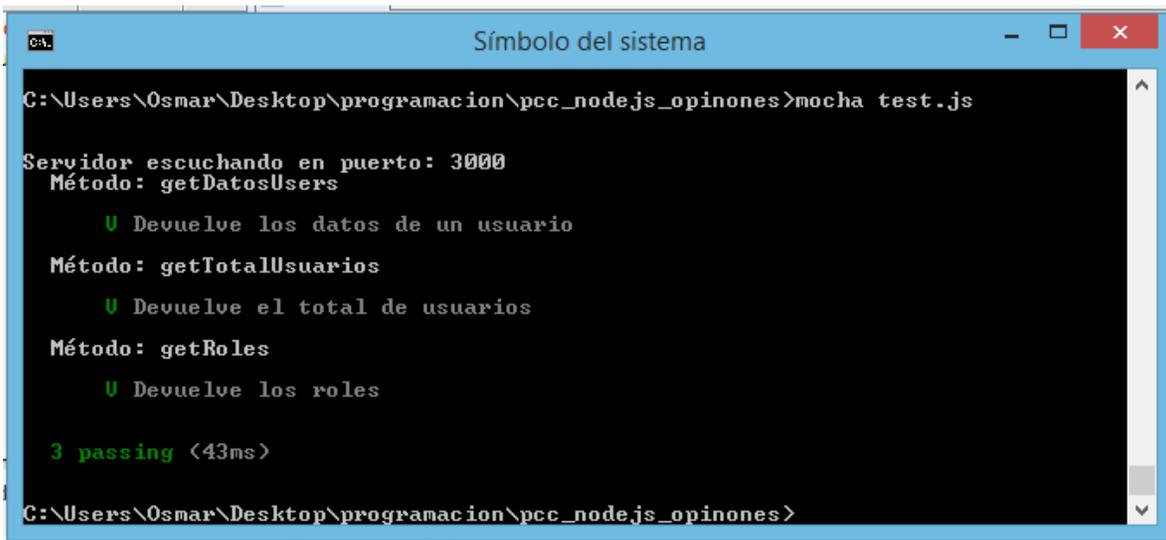
Estas pruebas se realizan para comprobar que el código funciona correctamente, haciendo uso de marco de pruebas, que ayudan en gran medida a los programadores automatizar las pruebas. Para el desarrollo de las pruebas unitarias se utilizó el marco de trabajo Mocha. A continuación se describe el mismo:

Mocha

Mocha es un marco de trabajo de JavaScript, contiene funciones que se ejecutan en Node.js o por medio de un navegador. Ejecuta las pruebas en series permitiendo reportes flexibles y exactos. Además facilita que el código sea óptimo una vez que se le haya realizado las pruebas

Mocha contiene sus propios comandos como por ejemplo: `--help` uno de los más importantes ya que muestra la ayuda de la herramienta. También `--version`, `--debug`, entre otros. Al realizar las pruebas muestra una V de color verde si al ejecutar el método devuelve el resultado esperado. Si el método falla muestra el resultado en color rojo. Las siguientes figuras muestran ejemplos de las pruebas unitarias realizadas en Mocha a las clases: Usuario, Opinión y Boletín.

Capítulo 3: Implementación y Prueba



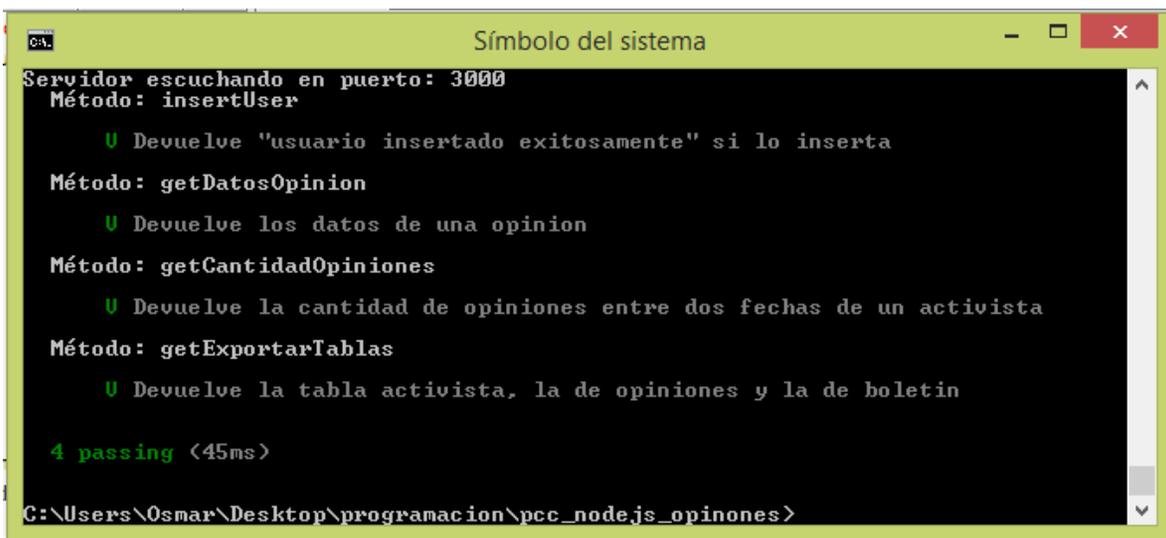
```
CA. Símbolo del sistema
C:\Users\Osmar\Desktop\programacion\pcc_nodejs_opinones>mocha test.js

Servidor escuchando en puerto: 3000
Método: getDatosUsers
    U Devuelve los datos de un usuario
Método: getTotalUsuarios
    U Devuelve el total de usuarios
Método: getRoles
    U Devuelve los roles

3 passing (43ms)

C:\Users\Osmar\Desktop\programacion\pcc_nodejs_opinones>
```

Figura 17 Prueba unitaria de la clase Usuario



```
CA. Símbolo del sistema
Servidor escuchando en puerto: 3000
Método: insertUser
    U Devuelve "usuario insertado exitosamente" si lo inserta
Método: getDatosOpinion
    U Devuelve los datos de una opinion
Método: getCantidadOpiniones
    U Devuelve la cantidad de opiniones entre dos fechas de un activista
Método: getExportarTablas
    U Devuelve la tabla activista, la de opiniones y la de boletin

4 passing (45ms)

C:\Users\Osmar\Desktop\programacion\pcc_nodejs_opinones>
```

Figura 18 Prueba unitaria de las clases Opinión y Boletín

Pruebas de aceptación

Las pruebas de aceptación son las especificaciones para el comportamiento deseado y la funcionalidad de un sistema. Describen, por una HU dada, cómo el sistema se encarga de ciertas condiciones e insumos y con qué tipo de resultados (Koskela, 2008).

Los clientes junto a un miembro del equipo de desarrollo son los encargados de revisar que los resultados de estas pruebas son los esperados. Una HU no se puede considerar terminada hasta que no pase los test de aceptación. Es recomendable publicar los

Capítulo 3: Implementación y Prueba

resultados de las pruebas de aceptación, para que todo el equipo de desarrollo esté al tanto de esta información.

A continuación se muestran los casos de pruebas de aceptación (CP), compuestos por los siguientes parámetros:

- **Código:** identificador del CP.
- **Historia de usuario:** nombre de la HU correspondiente al CP.
- **Nombre:** nombre de la funcionalidad que se le está realizando la prueba.
- **Descripción:** descripción de la prueba realizada.
- **Condiciones de ejecución:** condiciones necesarias para que se pueda realizar la prueba.
- **Pasos de ejecución:** describe los pasos a seguir para poder realizar la prueba.
- **Resultado esperado:** descripción de lo esperado después de realizada la prueba.
- **Evaluación de la prueba:** muestra si la prueba fue satisfactoria o insatisfactoria

Tabla 28 CP de Aceptación Registrar usuario

Caso de prueba de aceptación	
Código: HU1_CP1	Historia de usuario: 1
Nombre: Registrar usuario	
Descripción: Prueba para la funcionalidad registrar usuario.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador	
Pasos de ejecución: Para insertar los datos debe ir al menú principal y seleccionar la opción "Usuarios". En el submenú que se va a mostrar debe dar click en el link Registrar Usuario. Una vez introducido los datos se da clic en el botón "Registrar" y luego se muestra el usuario añadido a la base de datos.	
Resultado esperado: El usuario es registrado correctamente	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 29 CP Aceptación Autenticar usuario

Caso de prueba de aceptación	
Código: HU1_CP4	Historia de usuario: 1
Nombre: Autenticar usuario	
Descripción: Prueba para la funcionalidad autenticar usuario.	

Capítulo 3: Implementación y Prueba

Condiciones de ejecución. Los usuarios deben autenticarse ante el sistema.
Pasos de ejecución: Al correr el sistema se muestra una ventana donde el usuario debe registrarse con su usuario y contraseña para poder acceder al mismo.
Resultado esperado: El usuario es autenticado
Evaluación de la prueba: Prueba satisfactoria.

Tabla 30 CP de Aceptación Registrar opinión

Caso de prueba de aceptación	
Código: HU2_CP8	Historia de usuario: 2
Nombre: Registrar opinión	
Descripción: Prueba para la funcionalidad registrar opinión.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal.	
Pasos de ejecución: Para insertar una opinión debe ir al menú y seleccionar la opción "Municipio". En el submenú que se va a mostrar debe dar click en el link Registrar Opinión.	
Resultado esperado: La opinión es insertada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 31 CP de Aceptación Mostrar opinión

Caso de prueba de aceptación	
Código: HU2_CP11	Historia de usuario: 2
Nombre: Mostrar opinión	
Descripción: Prueba para la funcionalidad mostrar opinión.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o funcionario municipal	
Pasos de ejecución: Para mostrar una opinión deben dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en el link Mostrar opiniones. La búsqueda se puede filtrar por los datos tema, subtema, asunto, subasunto, red, sentido, boletín, activista, provincia, municipio, consejo popular, así como seleccionar el rango de fecha de la opinión que desea visualizar.	
Resultado esperado: Se muestra la opinión.	
Evaluación de la prueba: Prueba satisfactoria.	

Capítulo 3: Implementación y Prueba

Tabla 32 CP de Aceptación Registrar boletín

Caso de prueba de aceptación	
Código: HU3_CP12	Historia de usuario: 3
Nombre: Registrar boletín.	
Descripción: Prueba para la funcionalidad registrar boletín.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal.	
Pasos de ejecución: El usuario debe ir al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar debe dar click en el link Opciones para el control de boletines, donde aparece la sección Registrar Boletín en el que se debe de llenar los siguientes campos de textos: nombre del boletín y tipo de boletín, además de seleccionar la fecha en la que desea realizar la acción.	
Resultado esperado: Se registra el boletín.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 33 CP de Aceptación Mostrar boletín

Caso de prueba de aceptación	
Código: HU3_CP13	Historia de usuario: 3
Nombre: Mostrar boletín.	
Descripción: Prueba para la funcionalidad mostrar boletín.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal.	
Pasos de ejecución: Para mostrar un boletín el usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar debe dar click en el link Mostrar Boletines, en el que se le facilita la búsqueda a partir de los datos: nombre del boletín, tipo de boletín y la fecha del mismo, para luego ser visualizado.	
Resultado esperado: Se muestra el boletín.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 34 CP de Aceptación Registrar activista

Caso de prueba de aceptación	
Código: HU4_CP16	Historia de usuario: 4
Nombre: Registrar activista.	
Descripción: Prueba para la funcionalidad registrar activista.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal.	
Pasos de ejecución: Para registrar un activista el usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar debe dar click en el link Opciones para el control de activistas y dar clic en Registrar, donde llenará varios campos, dentro de ellos: nombre y apellidos, provincia, municipio y centro de trabajo.	
Resultado esperado: Se registra el activista en el sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Capítulo 3: Implementación y Prueba

Tabla 35 CP de Aceptación Mostrar activista

Caso de prueba de aceptación	
Código: HU4_CP17	Historia de usuario: 4
Nombre: Mostrar activista.	
Descripción: Prueba para la funcionalidad mostrar activista.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar debe dar click en el link Opciones para el control de activistas y dar clic en Mostrar activista, donde se le brinda la posibilidad de buscar el activista a partir de varios datos, dentro de ellos: nombre y apellidos, código, provincia, municipio	
Resultado esperado: Se muestra el activista.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 36 CP de Aceptación Exportar boletín

Caso de prueba de aceptación	
Código: HU7_CP21	Historia de usuario: 7
Nombre: Exportar boletín	
Descripción: Prueba para la funcionalidad exportar boletín	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o funcionario municipal.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en el link Opciones para el control de los boletines. Donde puede exportar el boletín de la fecha deseada, teniendo en cuenta los datos: opiniones, tema, subtema, asunto, subasunto, sentido, red, activista.	
Resultado esperado: Se exporta el boletín deseado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 37 CP de Aceptación Importar archivo

Caso de prueba de aceptación	
Código: HU8_CP22	Historia de usuario: 8
Nombre: Importar archivo.	
Descripción: Prueba para la funcionalidad importar archivo.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o especialista provincial.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Provincia". En el submenú que se va a mostrar deben dar click en la opción Importar archivo.	
Resultado esperado: Se importa el archivo deseado.	
Evaluación de la prueba: Prueba satisfactoria.	

Capítulo 3: Implementación y Prueba

Tabla 38 CP de Aceptación Crear opinión tipo

Caso de prueba de aceptación	
Código: HU9_CP23	Historia de usuario: 9
Nombre: Crear opinión tipo.	
Descripción: Prueba para la funcionalidad crear opinión tipo.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o especialista provincial.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Provincia". En el submenú que se va a mostrar deben dar click en la opción Mostrar Opiniones Tipos, en el que pueden crear una opinión tipo a partir de las opiniones registradas en el sistema.	
Resultado esperado: Se crea la opinión tipo.	
Evaluación de la prueba: Prueba satisfactoria.	

Se diseñaron un total de 26 casos de prueba, correspondientes a las pruebas de aceptación. En este epígrafe se muestran las más importantes relacionadas con las tareas de ingeniería realizadas en el epígrafe 3.1.1, los restantes se encuentran en el Anexo 4.

3.3.3 Ambiente de prueba

El ambiente donde fueron realizadas las pruebas estuvo compuesto fundamentalmente por computadoras destinadas a la codificación y análisis de las opiniones del Departamento de opinión pública del PCC de Cienfuegos.

Tabla 39 Ambiente de prueba de la propuesta de solución

Cantidad de PC	Sistema Operativo	RAM	Procesador
9	Windows XP	256 MB	Pentium Celerom 2.0 GHZ
1	Windows XP	1 GB	Dual Core 1.6 GHZ
1	Windows Seven	4 GB	I3 a 3.6 GHZ

3.3.4 Análisis de los resultados de las pruebas

Las pruebas se realizaron en 4 iteraciones que permitieron detectar un número pequeño de no conformidades y corregirlas. Como resultado se obtuvo que en la primera iteración se encontraron 10 NC de ellas 6 significativas y 4 no significativas, que fueron resueltas antes de pasar a la segunda iteración. En la segunda iteración se detectaron 8 NC de ellas 3 significativas y 5 no significativas. En la tercera iteración se encontraron 5 NC de ellas 1 significativa y 4 no significativas, y en la cuarta iteración 3 NC siendo las 3 no significativas.

Capítulo 3: Implementación y Prueba

Para hacer una evaluación más detallada de los resultados obtenidos se presenta el gráfico de la figura 19:

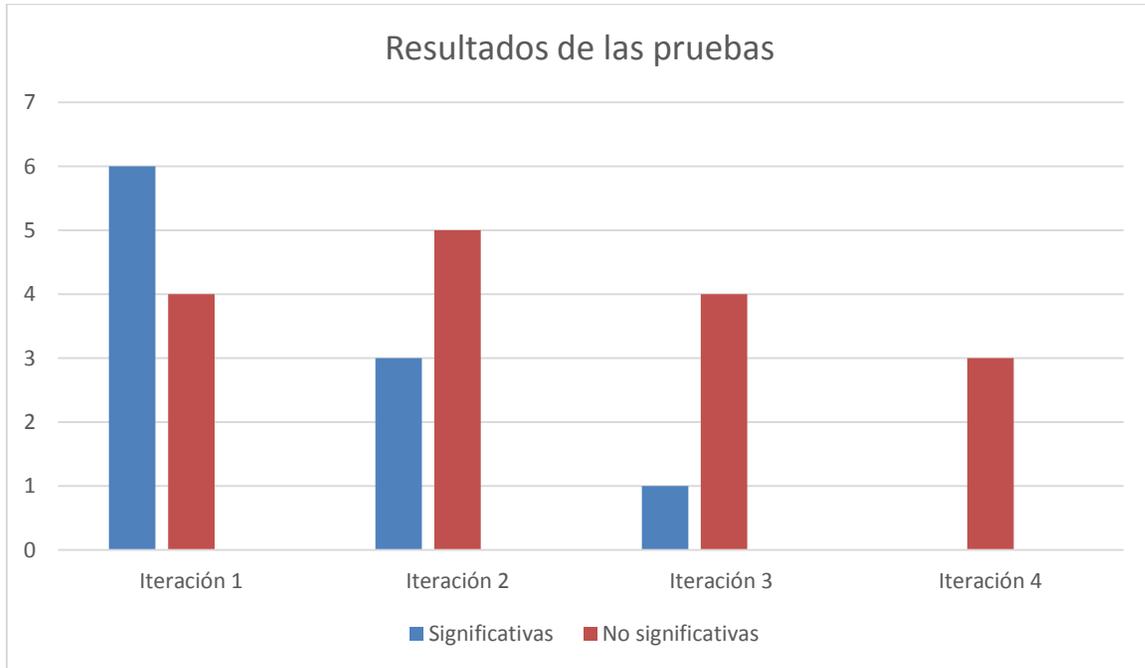


Figura 19 Resultados de las pruebas

Conclusiones del capítulo

Con el desarrollo del capítulo Implementación y prueba de la propuesta de solución se concluye lo siguiente:

- Los estándares de codificación utilizados permitieron organizar el código del sistema.
- La construcción del diagrama de despliegue propició mostrar la interacción de los componentes hardware y software de la aplicación final.
- El estudio de los niveles, tipos y métodos de prueba contribuyó a la confección de la estrategia de prueba, y comprobar el correcto funcionamiento de las funcionalidades.

Conclusiones generales:

Luego de haber desarrollado el Sistema de Gestión de información para el Departamento de opinión pública del PCC de Cienfuegos se concluye que:

- La investigación sobre el objeto de estudio y los métodos científicos utilizados permitieron identificar los conceptos y definiciones que respaldan la presente investigación.
- El análisis de las soluciones similares permitió conocer la existencia de herramientas diseñadas para la gestión de información de las opiniones públicas, también ayudó a comprender que no satisfacen las necesidades requeridas, por lo que es necesario la construcción de un sistema informático que mejore la codificación y análisis de las opiniones.
- La metodología seleccionada permitió generar los artefactos necesarios para obtener la aplicación, y satisfacer las necesidades del cliente como: Historia de usuarios, Plan de iteraciones, Plan de entregas, Tarjetas CRC y las Tareas de ingeniería.
- Las pruebas realizadas permitieron verificar el correcto funcionamiento de las funcionalidades definidas en el sistema.
- El sistema fue implementado en 3 municipios de Cienfuegos, Palmira, Cruce y Cienfuegos, quedando satisfecho con los resultados obtenidos, lo cual se hace constancia en la carta de aceptación entregada por el cliente.

Como resultado de la investigación se logró desarrollar un Sistema de Gestión de Información que cumple con los requisitos definidos y la satisfacción del cliente, mejorando los procesos de codificación y análisis de las opiniones.

Recomendaciones

Teniendo en cuenta los resultados obtenidos en la investigación se proponen las siguientes recomendaciones.

- Elaborar un Manual de usuario que guíe a los usuarios que interactúen con el sistema.
- Desarrollar la codificación automática de las opiniones haciendo uso de técnicas de Inteligencia Artificial.
- Desplegar el presente trabajo en las demás provincias del país.

Referencias

1. **Arévalo, Julio Alonso. 2007.** *Gestión de la Información, gestión de contenidos y conocimiento.* Salamanca : s.n., 2007.
2. **Beck, Kent y Fowler, Martin. 2000.** *Planning Xtreme Programming.* . s.l. : Addison Wesley, 2000. 1ra. Edición.
3. **CANÓS, Joseph, LETELIER, Patricio y PENADÉS, M. Carmen. 2003.** Metodologías Ágiles en el Desarrollo de Software. *Universidad Politécnica de Valencia.* [En línea] 2003. http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf.
4. **Castillo Oswaldo, Daniel Figueroa y Sevilla, Hector. 2008.** Fases de la Programación Extrema [online]. [En línea] 2008. <http://programacionextrema.tripod.com/fases.html>.
5. **Dashiell García Martínez, Luis Enrique Muñoz. 2011.** Repositorio Institucional. *Sistema de Procesamiento de Opiniones.* [En línea] Junio de 2011. http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_04950_11/1/TD_04950_11.pdf.
6. **Daza Díaz, Daniel Camilo y Florez Me, Alex Mauricio. 2013.** *DESARROLLO DE UNA ARQUITECTURA DE SOFTWARE PARA GESTIÓN DE INFORMACIÓN NO ESTRUCTURADA.* 2013.
7. **ExtremeProgramming. 2009.** Sitio Oficial ExtremeProgramming. *Extreme Programming.* [En línea] 2009. <http://www.extremeprogramming.org/>.
8. **IDE. 2009.** Slideshare. *Entornos-de-Desarrollos-Integrados.* [En línea] 2009. <http://es.slideshare.net/GhaBiithahh/entornos-de-desarrollo-integrados>.
9. **Ivar Jacobson, Grady Booch, James Rumbaugh.** El Proceso Unificado de Desarrollo de Software. [En línea] http://eva.uci.cu/file.php/158/Documentos/Bibliografia_general/Textos_Basicos/El_Proceso_Unificado_de_Desarrollo/01_Presentacion_y_contenido.pdf.
10. **Martinto, Revisado por MSc. Pedro Carlos Pérez.** El diseño metodológico de investigación científica. Teoría de muestreo: población y muestra. Diseño experimental y métodos. [En línea] http://eva.uci.cu/file.php/104/Tema_2/Recursos_bibliograficos/Diseno_metodol_de_la_invest-poblacion_y_muestra-_Metodos_y_diseno_experimental_-_Tema_3.pdf.

11. **Melissa Alperin, Katheen R. Miner. 1997.** Centers for Disease Control and Prevention. *Epi Info*. [En línea] Soquel, California: ToucandEd Publications, 1997. <http://wwwn.cdc.gov/epiinfo/>.
12. **MongoDB. 2003.** *MongoDB Documentation*. [En línea] 3 de Julio de 2003. <ftp://ucistore.uci.cu/software/Desarrollo/Base%20Datos/NoSQL%20DB/MongoDB/MongoDB-manual-master.pdf>.
13. **NetBeans. 2000.** Sitio Oficial NetBeans. *Features*. [En línea] diciembre de 2000. <http://netbeans.org/features/index.html>.
14. **Node.js. 2009.** Sitio Oficial Node.js. *Node.js on the road*. [En línea] 2009. <http://nodejs.org>.
15. **Olazabal, Isabel María Calás. 2009.** Repositorio Institucional. *Análisis y Diseño de una propuesta de aplicación Web para la gestión de opiniones y generación de boletines*. [En línea] Junio de 2009. http://repositorio_institucional.uci.cu/jspui/bitstream/ident/TD_2411_09/1/TD_2411_09.pdf.
16. **Paloma CÁCERES, MARCOS, Esperanza y KYBELE. 2001.** Procesos ágiles para el desarrollo de aplicaciones Web. *Taller de Web Engineering de las Jornadas de Ingeniería del Software y Bases de Datos*. [En línea] 2001. <http://dlsi.ua.es/webe01/articulos/s112.pdf>.
17. **Pérez, Javier Eguíluz.** Introducción a XHTML. [En línea] http://eva.uci.cu/file.php/106/Bibliografia_Basica/introduccion_xhtml.pdf.
18. —. JavaScript. [En línea] ftp://ftp.prod.uci.cu/PHP/Documentacion/Javascript/introducción_javascript.pdf.
19. **Pressman, Roger S.** *Software Engineering: a practitioner's approach, 7th ed.* 978-0-07-337597-7.
20. **Quiroga, Lourdes Aja. 2002.** Scielo. *Gestión de Información, gestión del conocimiento y gestión de la calidad en las organizaciones*. [En línea] Septiembre- Octubre de 2002. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352002000500004.
21. **RationalSoftware. 2005.** Rational Software. *Rational Unified Process Best Practices for Software Development Teams*. [En línea] 23 de Julio de 2005. http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf.
22. **Rolando Jesús Hidalgo, Rubiel A. González Labarta, Roniel López Álvarez.** *JavaScript SEAN Stack: Una Mirada al Futuro del Desarrollo Web*.

23. **Slideshare. 2008.** Slideshare. *Herramientas CASE*. [En línea] 2008. <http://es.slideshare.net/guestf131a9/herramientas-case>.
24. **SQLite. 2000.** Sitio Oficial SQLite. *About SQLite*. [En línea] 2000. <http://www.sqlite.org/about.html>.
25. **Stig Sæther Bakken, Alexander Aulbach, Egon Schmid, Jim Winstead, Lars Torben Wilson, Rasmus Lerdorf, Zeev Suraski, Andrei Zmievski, y Jouni Ahto.** Manual de PHP. [En línea] [ftp://ftp.prod.uci.cu/PHP/Documentacion/PHP/Manual de PHP.pdf](ftp://ftp.prod.uci.cu/PHP/Documentacion/PHP/Manual%20de%20PHP.pdf).
26. **Teixeira, Pedro.** *Professional Node.js*. s.l. : John Wiley & Sons, Inc., Indianapolis, Indiana.
27. **University of California, Santa Cruz. 2013.** Technology & Information Management (TIM). *What is Information Systems Management (ISM)?* [En línea] 2013. <http://tim.soe.ucsc.edu/undergraduates>.
28. **VisualParadigm.** Sitio Oficial Visual Paradigm. *Features*. [En línea] <http://www.visual-paradigm.com/features/>.

Anexos

Anexo 1 Historia de Usuarios

Tabla 40 HU Evaluar activista

Historia de Usuario	
Número: 5	Nombre: Evaluar activista
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Real: 1 semanas
Descripción: El sistema permitirá que el administrador y el funcionario municipal puedan evaluar a los activistas, a través de las opciones de filtrado en la que se selecciona el activista y el rango de fecha.	
Observaciones: El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal, el sistema le indicará los campos que son obligatorios para evaluar activista, además el sistema le mostrará la cantidad de opiniones y la sistematicidad de su entrega.	

Tabla 41 HU Mostrar evaluaciones

Historia de Usuario	
Número: 6	Nombre: Mostrar evaluaciones de los activistas.
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 2
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en desarrollo: Alto	Tiempo Real: 0.6 semanas
Descripción: El sistema permitirá que el administrador y el funcionario municipal puedan visualizar las evaluaciones de los activistas en un período de tiempo.	
Observaciones: El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal.	

Tabla 42 HU Asignar opinión tipo

Historia de Usuario	
Número: 10	Nombre: Asignar opinión tipo
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 4
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Real: 1 semanas
Descripción: El sistema permitirá que el administrador y el especialista provincial puedan asignar la categoría de opinión tipo a las opiniones registradas en el sistema.	
Observaciones: El usuario deberá estar autenticado con el rol administrador o el de especialista provincial.	

Tabla 43 HU Eliminar opinión tipo

Historia de Usuario	
Número: 11	Nombre: Eliminar opinión tipo
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 4
Prioridad: Alta	Tiempo Estimado: 2 días
Riesgo en desarrollo: Alto	Tiempo Real: 0.4 semanas
Descripción: El sistema permitirá que el administrador y el especialista provincial puedan eliminar una opinión dentro de la categoría opinión tipo	
Observaciones: El usuario deberá estar autenticado con el rol administrador o el de especialista provincial.	

Tabla 44 HU Mostrar opinión tipo

Historia de Usuario	
Número: 12	Nombre: Mostrar opiniones tipos
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 4
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en desarrollo: Alto	Tiempo Real: 0.6 semanas
Descripción: El sistema permitirá que el administrador, especialista provincial o el funcionario municipal puedan ver la cantidad de opiniones por tema, subtema, asunto, subasunto, red, tipo de boletín y activista.	
Observaciones: El usuario deberá estar autenticado con el rol administrador, especialista provincial o de funcionario municipal, el sistema les mostrará una tabla según el sentido, otra según el municipio y otra según sentido y municipio, donde el municipio varía según el consejo popular, si el usuario tiene rol de especialista provincial o de funcionario municipal respectivamente, además teniendo en cuenta un período de tiempo.	

Tabla 45 HU Mostrar cantidad de opiniones

Historia de Usuario	
Número: 13	Nombre: Mostrar cantidad de opiniones
Programador: Orlay Gainza Heredia, Osmar Núñez Rodríguez.	Iteración Asignada: 4
Prioridad: Alta	Tiempo Estimado: 5 días
Riesgo en desarrollo: Alto	Tiempo Real: 1 semanas
Descripción: El sistema permitirá que el administrador, especialista provincial o el funcionario municipal puedan ver la cantidad de opiniones por tema, subtema, asunto, subasunto, red, tipo de boletín y activista.	
Observaciones: El usuario deberá estar autenticado con el rol administrador, especialista provincial o de funcionario municipal, el sistema les mostrará una tabla según el sentido, otra según el municipio y otra según sentido y municipio, donde el municipio varí según el consejo popular, si el usuario tiene rol de especialista provincial o de funcionario municipal respectivamente, además teniendo en cuenta un período de tiempo.	

Anexo 2 Tarjetas CRC

Tabla 46 Tarjeta CRC Gestionar activista

Clase: Activista	
Responsabilidad	Colaboración
activista_controller.listadoEvaluar(filtro1)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.insertarEvaluacion(datos)	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.listadoEvaluaciones()	Crypto, Moment, SQLite3, Express, HTTP, Path
activista_controller.getDatosActivista(id)	Crypto, Moment, SQLite3, Express, HTTP, Path

Tabla 47 Tarjeta CRC Evaluar activista

Clase: Activista	
Responsabilidad	Colaboración
activista_controller.getEvaluacion()	Crypto, Moment, SQLite3, Express, HTTP, Path

Tabla 48 Tarjeta CRC Insertar opinión tipo

Clase: Opinión	
Responsabilidad	Colaboración
provincia_controller.insertarOpTipo(datos)	Crypto, Moment, SQLite3, Express, HTTP, Path

Tabla 49 Tarjeta CRC Asignar opinión tipo

Clase: Opinión	
Responsabilidad	Colaboración
provincia_controller.insertarAsignacion(datos)	Crypto, Moment, SQLite3, Express, HTTP, Path

Tabla 50 Tarjeta CRC Eliminar opinión tipo

Clase: Opinión	
Responsabilidad	Colaboración
provincia_controller.quitarOpinionTipo(op)	Crypto, Moment, SQLite3, Express, HTTP, Path

Tabla 51 Tarjeta CRC Mostrar cantidad de opiniones

Clase: Opinión	
Responsabilidad	Colaboración
opinion_controller.getOpiniones2(filtrado, fil_fecha)	Crypto, Moment, SQLite3, Express, HTTP, Path

Tabla 52 Tarjeta CRC Mostrar opinión tipo

Clase: Opinión	
Responsabilidad	Colaboración
provincia_controller.listadoOpinionesTipos()	Crypto, Moment, SQLite3, Express, HTTP, Path

Anexo 3 Interfaz de Usuario

Datos de los usuarios					
10	Num	Buscar <input type="text"/>			
Usuario	Nombre y apellidos	Fecha	Tipo usuario	Editar	Eliminar
admin	admin senior	05-22-2015	Administrador	Editar	Eliminar
orlay	Orlay Gainza	05-22-2015	Especialista provincial	Editar	Eliminar
malidia	Malidia Reinoso	05-22-2015	Funcionario municipal	Editar	Eliminar
Usuario	Nombre y apellidos	Fecha	Tipo usuario	Editar	Eliminar

1 de 3 en 3 Resultados < 1 >

Figura 20 Interfaz de usuario Mostrar usuarios

Registrar Opinión

Opinión *

Asunto *

Subasunto *

Sentido *

Descripción

Red *

Boletín *

Activista *

Registrar

Figura 21 Interfaz de usuario Registrar opinión

Registrar Boletín

Nombre del boletín: *

Tipo de boletín *

Fecha: * 

Registrar

Figura 22 interfaz de usuario Registrar boletín

Registrar Activista

Nombre y Apellidos: *

Código: *

Teléfono Celular:

Teléfono Casa:

Teléfono Trabajo:

Provincia: *

Municipio: *

Consejo Popular: *

Dirección: *

Centro de Trabajo: *

Sector: *

Militancia: SI No

Organización de Base: UJC PCC Ninguna

Encuestador: SI No

Comisión Científica: SI No

Supervisor: SI No

Especialista Equipo Opinión: SI No

Figura 23 Interfaz de usuario Registrar activista

Datos de los activistas

10 Num

Nombre y Apellidos	Código	# Celular	# Casa	# Trabajo	Provincia	Municipio	Consejo Popular	Dirección	Centro Trabajo	Sector	Militancia
No se arrojaron resultados											
Nombre y Apellidos	Código	# Celular	# Casa	# Trabajo	Provincia	Municipio	Consejo Popular	Dirección	Centro Trabajo	Sector	Militancia

0 de 0 en 0 Resultados

Figura 24 Interfaz de usuario Mostrar activistas

Cambiar contraseña

Usuario: * admin

Contraseña vieja: *

Contraseña nueva: *

repetir contraseña: *

Cambiar

Figura 25 Interfaz de usuario Cambiar contraseña

Opciones de filtrado

Activista Seleccione... Rango de fecha a

Clic para seleccionar

Datos de los activistas

Cantidad de Opiniones	Evaluar
Cantidad de Opiniones	Evaluar

Figura 26 Interfaz de usuario Opciones de filtrado

Mostrar Evaluaciones de los Activistas

10 Num Buscar

Código Activista	Fecha inicio	Fecha fin	Evaluación
No se arrojaron resultados			
Código Activista	Fecha inicio	Fecha fin	Evaluación

0 de 0 en 0 Resultados

Figura 27 Interfaz de usuario Mostrar evaluaciones de los activistas

Datos de las opiniones tipos

10 Num Buscar

Opinión	Tema	Subtema	Asunto	Subasunto	Sentido	Fecha	Descripción	Activista	Red	Boletín	Eliminar
No se arrojaron resultados											
Opinión	Tema	Subtema	Asunto	Subasunto	Sentido	Fecha	Descripción	Activista	Red	Boletín	Eliminar

0 de 0 en 0 Resultados

Figura 28 Interfaz de usuario Mostrar opiniones tipos

Anexo 4 Tareas de ingenierías

Tabla 53 TI Autenticar usuario

Tarea	
Número: 2	Número de HU: 1
Nombre: Autenticar usuario	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 17 de Enero 2015	Fecha fin: 18 de Enero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Autenticar usuario en el sistema.	

Tabla 54 TI Eliminar usuario

Tarea	
Número: 3	Número de HU: 1
Nombre: Eliminar usuario	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 14 de Enero 2015	Fecha fin: 15 de Enero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Eliminar usuario del sistema.	

Tabla 55 TI Editar usuario

Tarea	
Número: 5	Número de HU: 1
Nombre: Editar usuarios	
Tipo de Tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 22 de Enero 2015	Fecha fin: 24 de Enero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Editar usuarios.	

Tabla 56 TI Cambiar contraseña

Tarea	
Número: 6	Número de HU: 1
Nombre: Cambiar contraseña	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 25 de Enero 2015	Fecha fin: 26 de Enero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad cambiar contraseña de los usuarios.	

Tabla 57 TI Cerrar sesión

Tarea	
Número: 7	Número de HU: 1
Nombre: Cerrar sesión	
Tipo de Tarea: Desarrollo	Estimación: 1 días
Fecha de inicio: 28 de Enero 2015	Fecha fin: 28 de Enero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Cerrar sesión de los usuarios.	

Tabla 58 TI Editar opinión

Tarea	
Número: 9	Número de HU: 2
Nombre: Editar opinión	
Tipo de Tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 2 de Febrero 2015	Fecha fin: 4 de Febrero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Editar datos de una opinión.	

Tabla 59 TI Eliminar opinión

Tarea	
Número: 10	Número de HU: 2
Nombre: Eliminar opinión	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 6 de Febrero 2015	Fecha fin: 7 de Febrero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Eliminar opinión.	

Tabla 60 TI Editar boletín

Tarea	
Número: 14	Numero de HU: 3
Nombre: Editar boletín	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 18 de Febrero 2015	Fecha fin: 19 de Febrero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Modificar datos de un boletín.	

Tabla 61 TI Eliminar boletín

Tarea	
Número: 15	Numero de HU: 3
Nombre: Eliminar boletín	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 21 de Febrero 2015	Fecha fin: 22 de Febrero 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Eliminar un boletín.	

Tabla 62 TI Eliminar activista

Tarea	
Número: 18	Numero de HU: 4
Nombre: Eliminar activista	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 5 de Marzo 2015	Fecha fin: 6 de Marzo 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Eliminar activista.	

Tabla 63 TI Editar activista

Tarea	
Número: 19	Numero de HU: 4
Nombre: Editar activista	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 8 de Marzo 2015	Fecha fin: 9 de Marzo 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Editar activista.	

Tabla 64 TI Evaluar activista

Tarea	
Número: 20	Numero de HU: 5
Nombre: Evaluar activista	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 12 de Marzo 2015	Fecha fin: 13 de Marzo 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Evaluar activista.	

Tabla 65 TI Mostrar evaluaciones

Tarea	
Número: 21	Numero de HU: 6
Nombre: Mostrar evaluaciones	
Tipo de Tarea: Desarrollo	Estimación: 2 días
Fecha de inicio: 15 de Marzo 2015	Fecha fin: 16 de Marzo 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Mostrar evaluaciones de los activistas.	

Tabla 66 TI Crear opinión tipo

Tarea	
Número: 24	Numero de HU: 9
Nombre: Crear opinión tipo	
Tipo de Tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 5 de Abril 2015	Fecha fin: 8 de Abril 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Crear opinión tipo.	

Tabla 67 TI Asignar opinión tipo

Tarea	
Número: 25	Numero de HU: 10
Nombre: Asignar opinión tipo	
Tipo de Tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 12 de Abril 2015	Fecha fin: 14 de Abril 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Asignar opinión tipo.	

Tabla 68 TI Eliminar opinión tipo

Tarea	
Número: 26	Numero de HU: 11
Nombre: Eliminar opinión tipo	
Tipo de Tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 20 de Abril 2015	Fecha fin: 22 de Abril 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Eliminar una opinión de la categoría opinión tipo.	

Tabla 69 TI Mostrar opiniones tipo

Tarea	
Número: 27	Numero de HU: 12
Nombre: Mostrar opiniones tipo	
Tipo de Tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 1 de Mayo 2015	Fecha fin: 3 de Mayo 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Mostrar opiniones tipo.	

Tabla 70 TI Mostrar cantidad de opiniones

Tarea	
Número: 29	Numero de HU: 13
Nombre: Mostrar cantidad de opiniones	
Tipo de Tarea: Desarrollo	Estimación: 3 días
Fecha de inicio: 8 de Mayo 2015	Fecha fin: 10 de Mayo 2015
Programador responsable: Orlay Gainza Heredia, Osmar Núñez Rodríguez	
Descripción: Implementar la funcionalidad Mostrar cantidad de opiniones existentes en la base de datos.	

Anexo 5 Casos de pruebas de Aceptación

Tabla 71 CP de Aceptación Mostrar usuarios

Caso de prueba de Aceptación	
Código: HU1_CP3	Historia de usuario: 1
Nombre: Mostrar usuarios	
Descripción: Prueba para la funcionalidad Mostrar usuarios.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Usuarios". En el submenú que se va a mostrar debe dar click en el link Mostrar usuarios, donde se permite filtrar la búsqueda por diferentes datos: usuario, nombre y apellidos, email, tipo de usuario, fecha.	
Resultado esperado: Se muestran los usuarios registrados en el sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 72 CP de Aceptación Cambiar contraseña

Caso de prueba de Aceptación	
Código: HU1_CP5	Historia de usuario: 1
Nombre: Cambiar contraseña	
Descripción: Prueba para la funcionalidad Cambiar contraseña.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador, funcionario municipal o especialista provincial.	
Pasos de ejecución: Para cambiar la contraseña los usuarios deben dirigirse al menú principal y seleccionar la opción "Usuarios". En el submenú que se va a mostrar deben dar click en el link Cambiar contraseña, donde deben llenar los campos de textos: usuario, contraseña vieja, contraseña nueva y repetir contraseña.	
Resultado esperado: El usuario cambia su contraseña.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 73 CP de Aceptación Editar usuario

Caso de prueba de Aceptación	
Código: HU1_CP6	Historia de usuario: 1
Nombre: Editar usuario	
Descripción: Prueba para la funcionalidad Editar usuario.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Usuarios". En el submenú que se va a mostrar deben dar click en el link Mostrar usuarios. Luego selecciona Editar.	
Resultado esperado: Se modifican los datos de los usuarios.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 74 CP de Aceptación Eliminar opinión

Caso de prueba de Aceptación	
Código: HU1_CP2	Historia de usuario: 1
Nombre: Eliminar usuario	
Descripción: Prueba para la funcionalidad Eliminar usuario.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Usuarios". En el submenú que se va a mostrar debe dar click en el link Mostrar Usuarios, donde se le brinda la posibilidad de eliminar un usuario.	
Resultado esperado: El usuario es eliminado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 75 CP de Aceptación Cerrar sesión

Caso de prueba de Aceptación	
Código: HU1_CP7	Historia de usuario: 1
Nombre: Cerrar sesión	
Descripción: Prueba para la funcionalidad Cerrar sesión.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador, funcionario municipal o especialista provincial.	
Pasos de ejecución: Para cerrar sesión los usuarios deben dirigirse al menú situado en la parte superior derecha y seleccionar la opción "Salir".	
Resultado esperado: El usuario cierra su sesión.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 76 CP de Aceptación Editar datos de las opiniones

Caso de prueba de Aceptación	
Código: HU2_CP9	Historia de usuario: 2
Nombre: Editar datos de las opiniones	
Descripción: Prueba para la funcionalidad Editar datos de las opiniones.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol administrador o el de funcionario municipal.	
Pasos de ejecución: Para modificar datos de una opinión el usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar debe dar click en el link Mostrar Opiniones. La búsqueda se puede filtrar por los datos tema, subtema, asunto, subasunto, red, sentido, boletín, activista, provincia, municipio, consejo popular, además de seleccionar el rango de fecha de la opinión que desea modificar.	
Resultado esperado: La opinión es modificada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 77 CP de Aceptación Eliminar opinión

Caso de prueba de Aceptación	
Código: HU2_CP10	Historia de usuario: 2
Nombre: Eliminar opinión	
Descripción: Prueba para la funcionalidad Eliminar opinión.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de funcionario municipal y especialista provincial.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en el link Mostrar opiniones, donde buscar la opinión que desea eliminar.	
Resultado esperado: La opinión es eliminada.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 78 CP de Aceptación Editar boletín

Caso de prueba de Aceptación	
Código: HU3_CP14	Historia de usuario: 3
Nombre: Editar boletín	
Descripción: Prueba para la funcionalidad Editar boletín.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o funcionario municipal.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en Opciones para el control de los boletines y luego en el link Mostrar boletines en el que se le brinda la posibilidad de buscar el boletín y editar los datos: nombre del boletín, tipo de boletín y la fecha en que fue registrado.	
Resultado esperado: El boletín es modificado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 79 CP de Aceptación Eliminar boletín

Caso de prueba de Aceptación	
Código: HU3_CP15	Historia de usuario: 3
Nombre: Eliminar boletín	
Descripción: Prueba para la funcionalidad Eliminar boletín.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o funcionario municipal.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en el link Opciones para el control de los boletines, en el que se le brinda la posibilidad de eliminar el boletín deseado.	
Resultado esperado: El boletín es eliminado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 80 CP de Aceptación Eliminar activista

Caso de prueba de Aceptación	
Código: HU4_CP18	Historia de usuario: 4
Nombre: Eliminar activista	
Descripción: Prueba para la funcionalidad Eliminar activista.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o funcionario municipal.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en el link Opciones para el control de activistas, dar clic en mostrar activistas en el que se le brinda la posibilidad de eliminar un activista.	
Resultado esperado: El activista es eliminado.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 81 CP de Aceptación Editar activista

Caso de prueba de Aceptación	
Código: HU4_CP19	Historia de usuario: 4
Nombre: Editar activista	
Descripción: Prueba para la funcionalidad Editar activista.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o funcionario municipal.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en el link Opciones para el control de activistas, dar clic en mostrar activistas en el que se le brinda la posibilidad de modificar los datos de los mismos.	
Resultado esperado: El activista es modificado	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 82 CP de Aceptación Evaluar activista

Caso de prueba de Aceptación	
Código: HU5_CP20	Historia de usuario: 5
Nombre: Evaluar activista	
Descripción: Prueba para la funcionalidad Evaluar activista.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o funcionario municipal.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en el link Opciones para el control de activistas, luego dar clic en evaluar activista en el que se le brinda la posibilidad de modificar los datos de los mismos.	
Resultado esperado: El activista es modificado	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 83 CP de Aceptación de Mostrar evaluaciones

Caso de prueba de Aceptación	
Código: HU6_CP21	Historia de usuario: 6
Nombre: Mostrar evaluaciones	
Descripción: Prueba para la funcionalidad Evaluar activista.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o funcionario municipal.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en el link Opciones para el control de activistas, luego dar clic en mostrar evaluaciones en el que se le brinda la posibilidad de visualizarlas.	
Resultado esperado: Se muestran las evaluaciones.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 84 CP de Aceptación de Asignar opinión tipo

Caso de prueba de aceptación	
Código: HU10_CP24	Historia de usuario: 10
Nombre: Asignar opinión tipo.	
Descripción: Prueba para la funcionalidad Asignar opinión tipo.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o especialista provincial.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Provincia". En el submenú que se va a mostrar deben dar click en el link Mostrar Opiniones Tipos, donde pueden asignar a una opinión registrada en el sistema la categoría de opinión tipo.	
Resultado esperado: Se asigna a una opinión la condición de opinión tipo.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 85 CP de Aceptación Eliminar opinión tipo

Caso de prueba de Aceptación	
Código: HU11_CP25	Historia de usuario: 11
Nombre: Eliminar opinión tipo.	
Descripción: Prueba para la funcionalidad Eliminar opinión tipo.	
Condiciones de ejecución. El usuario deberá estar autenticado con el rol de administrador o especialista provincial.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Provincia". En el submenú que se va a mostrar deben dar click en el link Mostrar Opiniones Tipos, y eliminar la opinión con categoría tipo.	
Resultado esperado: Se elimina la opinión tipo.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 86 CP de Aceptación Mostrar cantidad de opiniones

Caso de prueba de Aceptación	
Código: HU12_CP26	Historia de usuario: 12
Nombre: Mostrar cantidad de opiniones	
Descripción: Prueba para la funcionalidad Mostrar cantidad de opiniones.	
Condiciones de ejecución: El usuario deberá estar autenticado con el rol de administrador, funcionario municipal o especialista provincial.	
Pasos de ejecución: El usuario debe dirigirse al menú principal y seleccionar la opción "Municipio". En el submenú que se va a mostrar deben dar click en el link Mostrar Opiniones, y visualizar la cantidad de opiniones registradas en el sistema.	
Resultado esperado: Se obtiene la cantidad de opiniones existentes en el sistema.	
Evaluación de la prueba: Prueba satisfactoria.	