

Implementación de una metodología de ingeniería de requerimientos en grandes proyectos de desarrollo de software

Diego Armando Valles Montalvo, Perfecto Malaquías Quintero Flores,
Higinio Nava Bautista

Tecnológico Nacional de México, Instituto Tecnológico de Apizaco,
Tlaxcala, México

armen0621@gmail.com, kmalakof@yahoo.fr, higinionava@hotmail.com

Resumen. Actualmente no se puede negar el gran crecimiento del mercado sobre los productos de software, es ahí donde surge el interés por asegurar el éxito de dichos productos y la calidad en su desempeño y algo muy importante que cumpla con las necesidades del cliente como usuario final. En este artículo se analizará el proceso de Ingeniería de requerimientos que muchas veces se da por alto en el proceso de desarrollo de software y su gran influencia en el éxito del software final. Es por ello que tras describir la metodología y siendo aplicada a un proyecto de desarrollo de software para la administración de patios de la industria automotriz, en el presente documento se discuten los resultados obtenidos en la implementación de una metodología de ingeniería de requerimientos en el inicio del proceso de desarrollo de un proyecto de software.

Palabras clave: Requerimientos, ingeniería de requerimientos (IR), ingeniería del software, industria automotriz.

Implementation of a Methodology of Requirements Engineering in Big Software Development Projects

Abstract. At present, it is not possible to deny the great growth of the market on software products, it is there where the interest arises to ensure the success of these products and the quality of their performance and something very important that meets the needs of the customer as an end user. This paper will analyze the requirements engineering process that is often overlooked in the software development process and its great influence on the success of the final software. It is for this reason that after describing the methodology and being applied to a software development project for the administration of patios of the automotive industry, the present document discusses the results obtained in the implementation of a methodology of requirements engineering in the beginning of the development process of a software project.

Keywords: Requirements, requirements engineering, automotive industry.

1. Introducción

Las actividades de diseño y construcción de software de computadoras son desafiantes, creativas y hasta divertidas. La construcción es tan irresistible que muchos desarrolladores de software quieren entrar en ella antes de comprender con claridad de qué es lo que se necesita.

La comprensión de los requerimientos de un problema está entre las tareas más difíciles que enfrenta un ingeniero de software. Cuando se piensa por primera vez acerca de ello, la Ingeniería de requerimientos no parece tan difícil.

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación, tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información, por lo tanto al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le llama ingeniería de requerimiento (IR).

1.1. Ingeniería del software

La Computer Society de la IEEE define la ingeniería del software como: “Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software, es decir, la aplicación de la ingeniería al software”.

La ingeniería del software es importante por dos razones para Sommerville [1], siendo éstas las siguientes:

1. Cada vez con mayor frecuencia, los individuos y la sociedad se apoyan en los avanzados sistemas de software. Por ende, se requiere producir económica y rápidamente sistemas confiables.
2. A menudo resulta más barato a largo plazo usar métodos y técnicas de ingeniería del software para los sistemas de software, que sólo diseñar los programas como si fuera un proyecto de programación personal. Para muchos tipos de sistemas, la mayoría de los costos consisten en cambiar el software después de ponerlo en operación.

Según Sommerville [1] explica que existen cuatro actividades fundamentales que son comunes a todos los procesos de software, y éstas son:

1. Especificación del software, donde clientes e ingenieros definen el software que se producirá y las restricciones en su operación.
2. Desarrollo del software, donde se diseña y programa el software.
3. Validación del software, donde se verifica el software para asegurar que sea lo que el cliente requiere.
4. Evolución del software, donde se modifica el software para reflejar los requerimientos cambiantes del cliente y del mercado.

Para Pressman [2] la Ingeniería del software es una tecnología estratificada, como se muestra en la Fig. 1, cualquier enfoque de la ingeniería debe ser sustentado en el compromiso con la calidad. La gestión de la calidad total, sigma seis y enfoques similares fomentan una cultura de mejora continua del proceso, y es esta cultura la que al final conduce al desarrollo de enfoques muy efectivos para la Ingeniería del software, la base que soporta la Ingeniería del software es un enfoque en la calidad.



Fig. 1. Estrados de la Ingeniería del software.

La base de la ingeniería del software es el estrado del proceso, el proceso define un marco de trabajo que debe establecerse para la entrega efectiva de la tecnología de la ingeniería del software. El proceso de software forma la base para el control de la gestión de los proyectos de software y establece el contexto en el cual se aplican los métodos técnicos, se generan los productos del trabajo, se establecen los fundamentos, se asegura la calidad.

Los métodos de la ingeniería del software proporcionan los “cómo” técnicos para construir software, los métodos abarcan un amplio espectro de tareas que incluyen la comunicación el análisis de requerimientos, el modelo del diseño, la construcción del programa, la realización de pruebas y soporte.

Las herramientas de la ingeniería del software proporcionan el soporte automatizado o semiautomatizado para el proceso y los métodos, cuando las herramientas se integran de forma que la información que cree una de ellas pueda usarla otra, se dice que se ha establecido un sistema para el soporte del desarrollo del software.

1.2. Requerimientos

El INTECO (Instituto Nacional de Tecnologías de la Comunicación) explica que los mejores productos, desde el punto de vista del usuario, son aquellos creados por desarrolladores que tienen muy claro lo que se pretende conseguir con el producto y cómo obtenerlo. Para llegar a este punto, se debe entender el trabajo del usuario, cómo afectará el producto a su trabajo y cómo se adecuará a los objetivos de la organización.

Un requerimiento es una característica del sistema o una descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer el propósito del sistema para

Pfleeger [3], lo que ha sido apropiadamente documentado y validado por el solicitante Berenbach [4]. Los requerimientos tratan exclusivamente sobre los fenómenos del dominio de aplicación y no sobre la máquina que los implementa Jackson [5].

Actualmente el mercado de productos de software ha incrementado considerablemente, es por ello que se debe de tener mayor atención análisis y su desarrollo para que así puedan cumplir con su propósito el cual es satisfacer las necesidades de un cliente. Para incrementar el grado de éxito se debe empezar con un buen levantamiento, análisis y diseño de requerimientos para que sirvan de base para los productos de software finales.

Hay dos formas de definir la Ingeniería de requerimientos según menciona Sommerville [1] y son:

1. Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones.
2. El proceso para establecer los servicios que el sistema debería proveer y las restricciones bajo las cuales debería operar y ser desarrollado.

1.3. Procesos de ingeniería de requerimientos

Los procesos de la Ingeniería de requerimientos incluyen cuatro actividades de alto nivel. Éstas se enfocan en valorar si el sistema es útil para la empresa (estudio de factibilidad), descubrir requerimientos (adquisición y análisis), convertir dichos requerimientos en alguna forma estándar (especificación) y comprobar que los requerimientos definan realmente el sistema que quiere el cliente (validación).

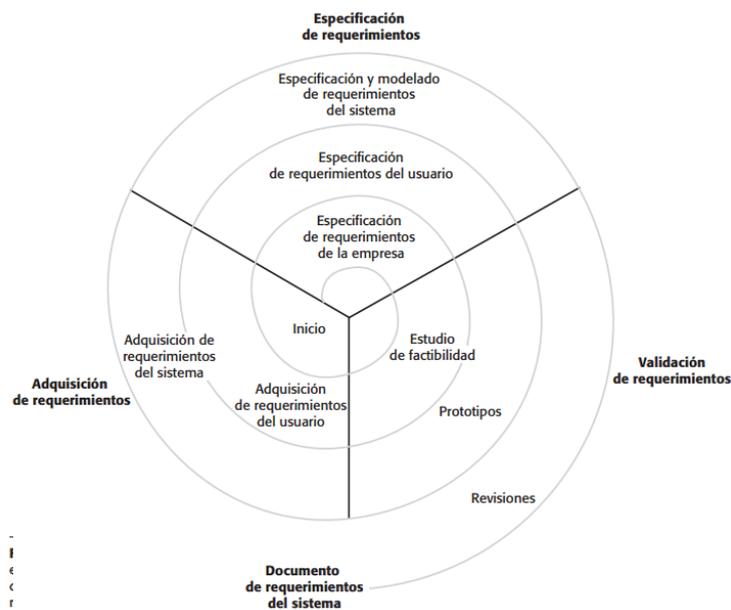


Fig. 2. Vista en espiral del proceso de Ingeniería de requerimientos.

La Fig. 2, presenta el entrelazamiento entre las diferentes fases del proceso de Ingeniería de requerimientos. Las actividades están organizadas como un proceso iterativo alrededor de una espiral, y la salida es un documento de requerimientos del sistema, la cantidad de tiempo y esfuerzo dedicados a cada actividad en cada iteración depende de la etapa del proceso global y el tipo de sistema que está siendo desarrollado. En el inicio del proceso, se empleará más esfuerzo para comprender los requerimientos empresariales de alto nivel y los no funcionales, así como los requerimientos del usuario para el sistema. Más adelante en el proceso, en los anillos exteriores de la espiral, se dedicará más esfuerzo a la adquisición y comprensión de los requerimientos detallados del sistema.

1.4. Adquisición y análisis de requerimientos

Sommerville [1] dice que después de un estudio de factibilidad inicial, la siguiente etapa del proceso de Ingeniería de requerimientos es la adquisición y el análisis de requerimientos. En esta actividad, los ingenieros de software trabajan con clientes y usuarios finales del sistema para descubrir el dominio de aplicación, qué servicios debe proporcionar el sistema, el desempeño requerido de éste, las restricciones de hardware, etcétera.

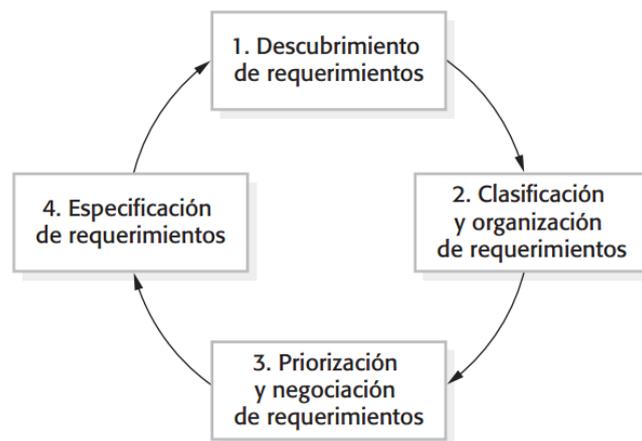


Fig. 3. El proceso de adquisición y análisis de requerimientos.

En la Fig. 3, se muestra un modelo del proceso de adquisición y análisis. Cada organización tendrá su versión o ejemplificación de este modelo general, dependiendo de factores locales, tales como experiencia del personal, tipo de sistema a desarrollar, estándares usados, etcétera.

Las actividades del proceso son:

- Descubrimiento de requerimientos. Éste es el proceso de interactuar con los participantes del sistema para descubrir sus requerimientos. También los

requerimientos de dominio de los participantes y la documentación se descubren durante esta actividad.

- Clasificación y organización de requerimientos. Esta actividad toma la compilación no estructurada de requerimientos, agrupa requerimientos relacionados y los organiza en grupos coherentes. La forma más común de agrupar requerimientos es usar un modelo de la arquitectura del sistema, para identificar subsistemas y asociar los requerimientos con cada subsistema. En la práctica, la Ingeniería de requerimientos y el diseño arquitectónico no son actividades separadas completamente.
- Priorización y negociación de requerimientos. Inevitablemente, cuando intervienen diversos participantes, los requerimientos entrarán en conflicto. Esta actividad se preocupa por priorizar los requerimientos, así como por encontrar y resolver conflictos de requerimientos mediante la negociación. Por lo general, los participantes tienen que reunirse para resolver las diferencias y estar de acuerdo con el compromiso de los requerimientos.
- Especificación de requerimientos. Los requerimientos se documentan e ingresan en la siguiente ronda de la espiral. Pueden producirse documentos de requerimientos formales o informales.

1.5. Validación de los requerimientos

La validación de requerimientos es el proceso de verificar que los requerimientos definan realmente el sistema que en verdad quiere el cliente. Se traslapa con el análisis, ya que se interesa por encontrar problemas con los requerimientos. La validación de requerimientos es importante porque los errores en un documento de requerimientos pueden conducir a grandes costos por tener que rehacer, cuando dichos problemas se descubren durante el desarrollo del sistema o después de que éste se halla en servicio Sommerville [1].

Hay algunas técnicas de validación de requerimientos que se usan individualmente o en conjunto con otras:

- Revisiones de requerimientos. Los requerimientos se analizan sistemáticamente usando un equipo de revisores que verifican errores e inconsistencias.
- Creación de prototipos. En esta aproximación a la validación, se muestra un modelo ejecutable del sistema en cuestión a los usuarios finales y clientes. Así, ellos podrán experimentar con este modelo para constatar si cubre sus necesidades reales.
- Generación de casos de prueba. Los requerimientos deben ser comprobables. Si las pruebas para los requerimientos se diseñan como parte del proceso de validación, esto revela con frecuencia problemas en los requerimientos. Si una prueba es difícil o imposible de diseñar, esto generalmente significa que los requerimientos serán difíciles de implementar, por lo que deberían reconsiderarse.

1.6. Administración de los requerimientos

Los requerimientos para los grandes sistemas de software siempre cambian. Una razón es que dichos sistemas se desarrollaron por lo general para resolver problemas “complejos”: aquellos problemas que no se pueden definir por completo. Como el problema no se logra definir por completo, los requerimientos del software están condenados también a estar incompletos. Durante el proceso de software, la comprensión que los participantes tienen de los problemas cambia constantemente ver Fig. 4. Entonces, los requerimientos del sistema también deben evolucionar para reflejar esa visión cambiante del problema.

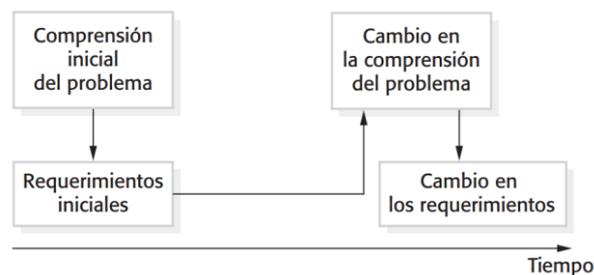


Fig. 4. El proceso de adquisición y análisis de requerimientos.

2. Estado del arte

La ingeniería de requerimientos es la fase más eficaz del proceso de desarrollo de software. Su objetivo es recoger buenos requerimientos de los interesados en la forma correcta. Es importante por cada organización para desarrollar productos de software de calidad que puede satisfacer las necesidades del usuario.

La ingeniería de requerimientos para el proceso de desarrollo de software es un ejercicio complejo que considera producto demandas de un gran número de puntos de vista, papeles, responsabilidades y objetivos. Por lo tanto, se hace necesario aplicar prácticas de ingeniería requerimiento en todas las fases de proceso de desarrollo de software Dhirendra [6].

A. Charan Kumari [7] hace mención que en la fase de ingeniería requerimientos del ciclo de vida de desarrollo de software, uno de las principales preocupaciones de los ingenieros de software es seleccionar un conjunto de requerimientos de software para su implementación en la próxima versión del software de muchos requerimientos propuestos por los clientes, mientras que el equilibrio del presupuesto y la satisfacción del cliente.

El autor Philipp Holtkamp [8] presenta que el enfoque más común es escribir los requerimientos de usuario mediante el lenguaje natural. La ventaja es que el lenguaje natural es el principal medio de comunicación entre las partes interesadas. Sin embargo, problemas como la imprecisión, las incomprensiones, la ambigüedad y la incoherencia son comunes cuando se utiliza el lenguaje natural.

Sin embargo, el lenguaje natural no es estructurado ni formal ni gráfico, y puede ser demasiado orientado a los algoritmos y lenguajes de programación concretos. Un conocido diagrama utilizado para el modelado de requerimientos son los casos de uso. Incluso antes de UML surgió como el principal lenguaje de modelado de la ingeniería del software.

El autor Michel dos Santos Soares [9] discute que el modelado debe proporcionar los medios para expresar las necesidades gráficas. Los modelos gráficos comunes pueden facilitar la comunicación de los modelos para los interesados. Los modelos deben ser legible para el ser humano, como los múltiples actores involucrados tienen que comprender los modelos. En este caso, el equilibrio es necesario, como por ejemplo los requerimientos más legibles por la máquina, menos legible se vuelven para el ojo humano.

3. Implementación de la metodología de ingeniería de requerimientos

Para la adquisición de requerimientos se implementaron cuestionarios y entrevistas como herramientas de recolección de datos de información como lo menciona la metodología, tras la obtención de la información se hizo el análisis de la misma para la generación de requerimientos de software, por medio de la clasificación de requerimientos funcionales y no funcionales.

Se definió una nomenclatura para la clasificación de los requerimientos una vez definidos y priorizados por el nivel de importancia e impacto en el desarrollo del sistema, dejando así definidas las relaciones entre requerimientos y separando los requerimientos que son determinados como de menor impacto. Al realizar la clasificación definimos en una escala del 1 al 10 la importancia y su grado de dificultad al ser desarrollado, y así determinar el tiempo que tomara para ser concluido.

Logrando así la obtención de los primeros requerimientos informales para su negociación con el cliente disipando diferencias negativas para futuras revisiones.

Mediante una reunión de revisión de avance se validaron los requerimientos definidos por todo el equipo de trabajo y el cliente, por medio de esto se puede verificar si lo propuesto satisface con las necesidades del cliente recolectadas inicialmente en el proyecto. Se logró encontrar errores de análisis y priorización evitando así costos innecesarios en el futuro de proyecto.

Para una mejor representación de los requerimientos establecidos se elaboraron los prototipos de bajo, medio y alto nivel para la representación gráfica de los requerimientos y llevar al cliente a una conceptualización más sólida sobre el análisis que se realizó al problema propuesto.

A todo este proceso se elaboraron los documentos que sustenten los cambios y las entregas que se realizaron, que sirven de apoyo a futuras explicaciones y para dejar como acuerdos de negociación evitando así futuros mal entendidos y teniendo documentación base para el desarrollo del sistema.

4. Resultados experimentales

Tomando como caso experimental el desarrollo de un proyecto de software para la “Administración de patios de la industria automotriz”, se implementó el proceso que propone la Ingeniería de requerimientos, y se obtuvieron los siguientes resultados:

4.1. Especificación de requerimiento de software (ERS)

En la Fig. 5, se muestra un extracto del documento ERS donde se describen los requerimientos ordenándolos de manera secuencial y por jerarquías de dependencias, logrando así su interpretación a la hora de realizar el diseño y codificación del software.

Una de las tareas más importante de la ingeniería de requerimientos es la recolección y análisis de los requerimientos, es una tarea ardua entre una serie de métodos para la recolección es por ello que nos dimos a la tarea de aplicar entrevistas y sesiones de pláticas con el cliente para así juntar la información necesaria y tras un análisis elaborar un documento formal como lo es el ERS que sirve como evidencia de conformidad por parte del cliente.

4. Requerimientos funcionales para la aplicación en escritorio

4.1. Perfiles de usuario

El sistema contará con los siguientes perfiles de usuario:

4.1.1. Administrador general

- Persona encargada de administrar empresas dentro del sistema

4.1.2. Administrador de la empresa

- Persona encargada de controlar, gestionar y conceder permisos a los demás usuarios.

4.1.3. Usuario

- Persona que accede al sistema con un perfil y permisos definidos por el administrador de la empresa a la que pertenezca.

4.2. Acceso al sistema

4.2.1. El proceso será a través de un navegador web, el cual deberá tener una interfaz de inicio de sesión.

4.2.2. En esta interfaz el usuario tendrá las siguientes opciones:

4.2.3. Iniciar sesión

4.2.3.1. El usuario podrá acceder al sistema a través de un correo electrónico y contraseña.

4.2.3.2. Si no ingresa usuario o contraseña, el sistema mostrará un mensaje informativo solicitando la información y no permitirá el acceso al sistema.

4.2.3.3. En caso de que el correo y/o contraseña no se encuentren registrados, el sistema mostrará un mensaje informativo.

4.2.3.4. El sistema validará si los datos del correo electrónico y la contraseña son correctos, de no ser así mostrará un mensaje informativo.

4.2.3.5. Si la cuenta se encuentra inactiva se mostrará un mensaje informativo.

4.2.3.6. Si el sistema valido el correo electrónico y la contraseña, se dice que el usuario inicia sesión de trabajo y el sistema mostrará la interfaz de bienvenida.

Fig. 5. Especificación de Requerimientos de Software.

4.2. Diagrama de casos de uso

Un análisis de funcionalidad como primera instancia del análisis de los requerimientos da como resultado la elaboración de los casos de uso, para conocer así las principales interacciones que tendrá el usuario con el software, ver Fig. 7.



ID - UC:	Caso de uso 05
Nombre de UC:	Registro de daños

Actores:	Administrador de empresa y/o usuario del sistema
Descripción:	En éste módulo se permite administrar los daños que se ..

Fig. 7. Diagrama de casos de uso.

Los casos de uso sienten uno de los diagramas típicos que se elaboran tras iniciar el análisis de los requerimientos, quedo reflejado en este proyecto ya que podemos discernir las interacciones del usuario así como los posibles escenarios en los cuales se puede encontrar el usuario dentro del software.

4.3. Diagrama de actividades

Los diagramas de actividades nos permitieron ver reflejada de manera el flujo de las actividades del usuario dentro del producto de software, ver Fig. 8. A grandes rasgos el usuario realiza una serie de actividades dentro de software las cuales se reflejaron en la elaboración de diagramas de secuencia, para realizar de manera visual las posibles actividades que el usuario realizara y mostrar así un flujo de acciones.

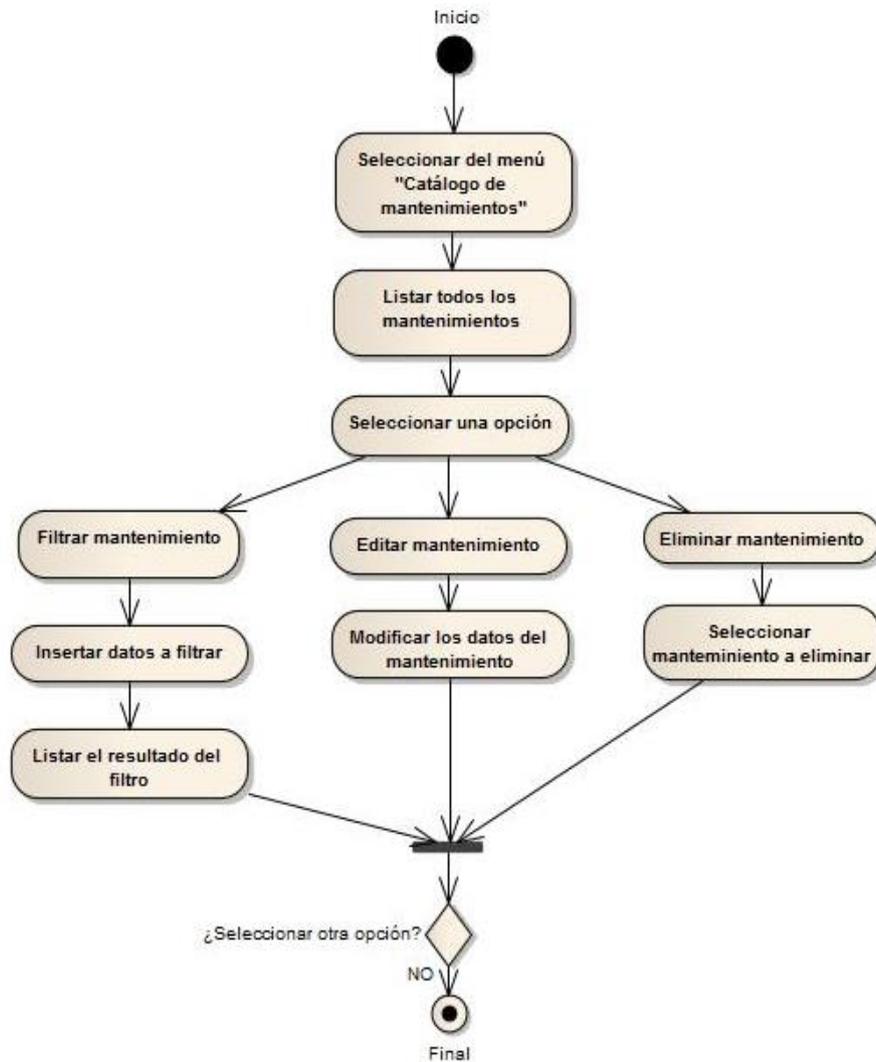


Fig. 8. Diagrama de actividades.

4.4. Diagrama de procesos BPM

En la Fig. 9, vemos el resultado de un diagrama BPM que sirve para ejemplificar el proceso de interacción de un usuario dentro del producto de software, permitiendo así la retroalimentación entre usuario y analista.

El flujo de procesos que refleja el diagrama de procesos BPM, nos da un marco referencial más exacto del funcionamiento general del software, en este caso se dividió por módulos de acciones y a su vez en submódulos, y por como resultado se obtuvo un análisis más a fondo asegurando la viabilidad del proyecto.

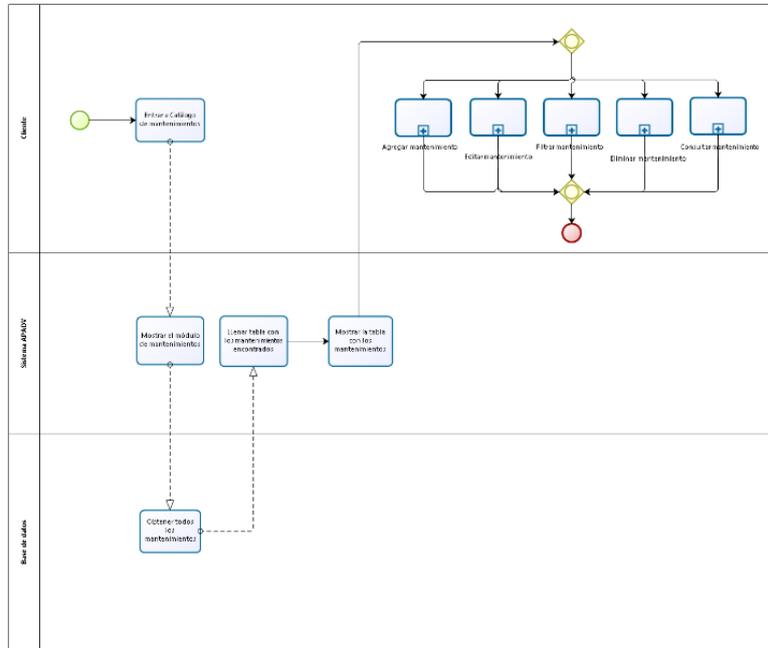


Fig. 9. Diagrama de procesos BPM.

4.5. Prototipos

Actualmente la creación de prototipos hacen más fácil la tarea de recolección de requerimientos asegurando que lo entendido hasta el momento por el analista es lo que realmente el usuario necesita, ver Fig. 10.

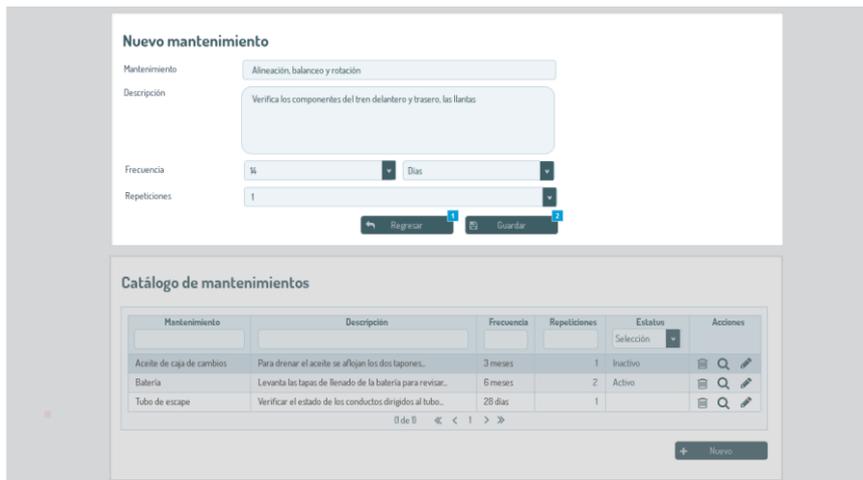


Fig. 10. Prototipo de alto nivel.

Los prototipos elaborados fueron un plus total al proyecto generando mayor aceptación por parte del cliente el cual de manera inmediata pudo ver reflejada las respuestas a su lista de necesidades.

5. Conclusiones y trabajo futuro

Se concluye que la implementación adecuada de la ingeniería de requerimientos al inicio de proyectos para la elaboración de software nos permite identificar y entender de manera correcta las necesidades de los clientes, dando como resultado la elaboración del software a la medida de sus requerimientos, y con esto aseguramos el éxito de la aceptación del producto de software por parte de usuario final.

Para las investigaciones futuras, se propone el diseño de un nuevo proceso de Ingeniería de requerimientos rescatando las actividades de mayor impacto y proponiendo nuevas actividades dentro del proceso, aportando así un proceso funcional para proyectos de software generales.

Agradecimientos. Queremos agradecer a todos los impulsores que trabajan a favor de mejorar el desarrollo de software.

Referencias

1. Sommerville, I.: Ingeniería del software. Estado de México, Pearson (2011)
2. Pressman, R. S.: Ingeniería del software. Un enfoque práctico. México, MacGraw Hill (2005)
3. Pfleeger, S. L.: Software Engineering: Theory and Practice. NJ, USA, Upper Saddle River (2001)
4. Brian, D. P.: Software & Systems Requirements Engineering: In Practice. Inc., New York, NY, USA, McGraw-Hill (2009)
5. Jackson, M.: Software requirements & specifications: a lexicon of practice, principles and prejudices. New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (1995)
6. Dhirendra, U. S.: An Effective Requirement Engineering Process Model for Software Development and Requirements Management. In: International Conference on Advances in Recent Technologies in Communication and Computing (2010)
7. Charan, K. S.: Comparing the performance of quantum-inspired evolutionary algorithms for the solution of software requirements selection problem. Elsevier (2016)
8. Holtkamp, P.: Soft competency requirements in requirements engineering, software design, implementation and testing. Elsevier (2014)
9. dos Santos, M, Vrancken, J.: User Requirements modeling and analysis of software-intensive systems. Elsevier (2010)