

# CASO PRÁCTICO

**Nombre y apellidos:** Wolfgang Oswaldo Mora Celi

**Fecha:** 16/01/2024

## El levantamiento de Requisitos de Software

El inicio de toda la actividad de desarrollo de aplicaciones comienza por el levantamiento de requisitos, estando este proceso repetido en cada nueva iteración del proyecto de software.

Sommerville (2003) propone un proceso genérico del levantamiento y análisis que contiene las siguientes actividades:

**Comprensión de dominio:** Los analistas deben desarrollar su comprensión del dominio de aplicación.

**Recolección requisitos:** Es el proceso de interactuar con los diferentes actores del sistema para descubrir sus necesidades. La comprensión del dominio se desarrolla más durante esta actividad.

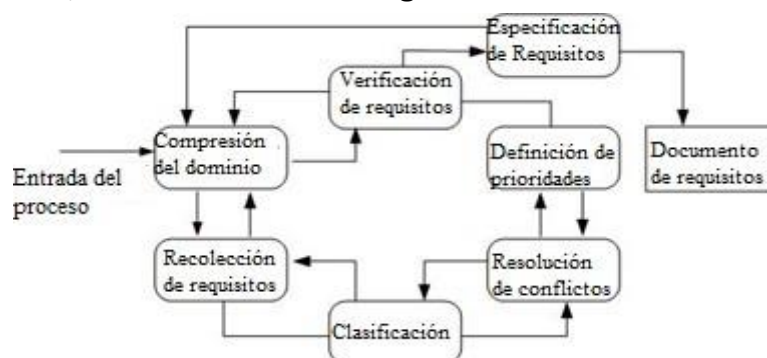
**Clasificación:** Esta actividad considera todos los requisitos no estructurados y los organiza en grupos coherentes.

**Resolución de conflictos:** Cuando varios actores están involucrados, los requisitos colectados pueden conducir a conflictos. Esta actividad tiene como objetivo resolver estos conflictos;

**El establecimiento de prioridades:** En cualquier conjunto de requisitos, algunos son más importantes que otros. Esta etapa implica la interacción con las partes interesadas para definir los requisitos más importantes;

**Verificación de Requisitos:** Los requisitos son comprobados para averiguar si están completos y coherentes y están en consonancia con lo que los actores quieren del sistema.

El estudio y análisis de los requisitos es un proceso iterativo, con una validación continua de una actividad a otra, como se muestra en la Figura 1.



**Figura 1:** Proceso de levantamiento y análisis de requisitos (SOMMERVILLE, 2003).

---

## ***Instrucciones para el desarrollo de la actividad***

---

### **I. Con base a la descripción anterior, responda a las siguientes preguntas:**

#### **1. ¿Cuáles son las principales dificultades encontradas en la fase de levantamiento de requisitos?**

La fase de levantamiento de los requisitos se vuelve la fase más importante a la hora de desarrollar un software pues este permite tener una idea clara de lo que va a hacer el software, funcionamiento y usabilidad que este tendrá pues “es el proceso por el cual se transforman los requerimientos declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones”. (Guide, 1987).

Dada su gran importancia es que este llega a tener múltiples dificultades a la hora de aplicar esta fase, estas dificultades llegan a ser tanto de ámbito humano como la dificultad de la que suponga la naturaleza del software a desarrollar, las principales dificultades son:

- **Comunicación insuficiente:** La falta de comunicación clara y efectiva es un detonante de problemas en equipos de desarrollo y clientes, esto supone equivocaciones y un tiempo de desarrollo muy extenso.
- **Cambios en los requisitos:** Los requisitos pueden cambiar a lo largo del tiempo ya sea a cambios en el entorno empresarial o por nuevas ideas del cliente, esto detona en múltiples cambios mientras se realiza un método específico.
- **Falta de participación de los interesados:** La falta de involucramiento activo de los usuarios puede llevar a una comprensión inadecuada de los requisitos del sistema dando problemas futuros a la hora de desarrollar el software de manera incorrecta.
- **Complejidad del sistema:** Si el sistema es complejo, puede ser difícil identificar y documentar todos los requisitos de manera precisa y completa.
- **Expectativas no gestionadas:** Las expectativas poco realistas o mal gestionadas por parte de los usuarios finales pueden llevar a la insatisfacción con el producto final.

#### **2. ¿Cuáles son las principales técnicas para el levantamiento de requisitos? Explique brevemente cada una de las técnicas.**

El levantamiento de requisitos al ser una fase tan importante para el desarrollo, ha sido un gran tema de interés para múltiples autores que indican las técnicas más importantes para realizar esto, sin embargo, estas coinciden muchas veces en las mismas técnicas. Las más comunes son:

- Entrevistas: Tener una comunicación directa con los usuarios finales, clientes y otros interesados permite tener información valiosa acerca de las necesidades y expectativas que tendrá el software.
- Recolección requisitos: Es el proceso de interactuar con los diferentes actores del sistema para descubrir sus necesidades. La comprensión del dominio se desarrolla más durante esta actividad.
- Comprensión de dominio: Los analistas deben ser capaces de desarrollar una comprensión del dominio de la aplicación a desarrollar según la información del usuario.
- Resolución de conflictos: Cuando varios actores están involucrados, los requisitos colectados pueden conducir a conflictos, estos deben ser identificados de manera temprana para su resolución inmediata.
- Historias de usuario: Describir las funcionalidades desde la perspectiva que tendrá el usuario, esto permitirá entender de mejor manera sus necesidades.
- Casos de uso: Desarrollar escenarios detallados que describan la interacción de los usuarios con el sistema, permitiendo de esta forma entender mejor los requisitos.
- Clasificación: Esta actividad se consideran todos los requisitos no estructurados y los organiza en grupos coherentes.
- Prototipado: Crear prototipos del sistema que permitan a los usuarios interactuar y proporcionar retroalimentación temprana.
- Verificación de Requisitos: Los requisitos son comprobados para averiguar si están completos y coherentes y están en consonancia con lo que los actores quieren del sistema.

**3. JAD (Joint Aplicación Diseño) es una metodología que permite la extracción de información de alta calidad de usuarios en un corto período de tiempo a través de reuniones estructuradas que buscan decisiones por consenso, que es una de las formas más productivas de decisiones en grupo. Hacer una búsqueda en Internet y enviar sus principios básicos y sus principales etapas.**

El diseño conjunto de aplicaciones (JAD) es un término genérico que describe una variedad de métodos patentados y desarrollados a medida para realizar talleres en los que usuarios y desarrolladores técnicos trabajan juntos en la planificación de proyectos de sistemas de información (IS) (E. Davidson 1999). Es decir esta metodología busca que exista una interacción constante entre los usuarios finales y los desarrolladores, esto tiene la finalidad de mejorar la calidad del software en la forma en como se obtiene la información relevante de la necesidades, requisitos y funcionalidades que deberá tener un sistema informático.

Esta metodología es usada generalmente en proyectos en los cuales los requisitos del sistema sean difíciles de entender o la complejidad del algoritmo sea alta y sea necesario tener una comprensión precisa de los requisitos que tiene el usuario. Se dice que JAD conduce a una mayor calidad, menores costos y una reducción del

tiempo del ciclo de vida (Carmel 1993).

JAD puede verse como una técnica y una metodología. Es una técnica porque es una estructura para realizar una reunión de diseño con los usuarios participantes. Es una metodología porque cuando se introducen en el SDLC(Ciclo de desarrollo del software), las reuniones de JAD forman el núcleo en torno al cual giran todas las actividades. La metodología JAD enfatiza la estructura y la agenda (Carmel 1993). JAD permite facilitar la comunicación efectiva y asegura que los requisitos del sistema estén alineados con las necesidades reales del negocio.

#### **Bibliografía**

**Guide, STARTS. 1987. Ingenieria de Requerimientos.**

**E. Davidson. 1999 Joint application design (JAD) in practice.**

**Carmel, Erran, Randall and Joey. 1993. PD and Joint Application Design: A Transatlantic Comparison**

- II. Elabore su respuesta a estas interrogantes basándose en el estudio de los materiales de la asignatura y otras fuentes que considere oportuno consultar.**
- III. Una vez completadas las respuestas, haga entrega del documento a través del icono de la actividad.**