



**Universidad Nacional  
Autónoma de México**



**Facultad de ingeniería**

**Documentación del tutorial.**

Ingeniería en software

**Grupo: 4**

**Equipo: 5**

**Semestre: 2026 – 1**

**Profesor: Orlando Zaldívar Zamorategui**

**Fecha de entrega:**

**13 de Noviembre 2025**

**Líder del equipo: Domínguez Ruiz Samantha Atziry -  
322273672**

# **Temario: Fundamentos y Técnicas de Ingeniería de Requisitos de Software**

## **Módulo 1: Introducción a los Requerimientos de Software**

### **1.1. Objetivo de Aprendizaje**

### **1.2. Planteamiento del Problema**

## **Módulo 2: Fundamentos de los Requerimientos**

### **2.1. Propósitos y Beneficios Clave**

### **2.2. Clasificación de Requerimientos**

#### **2.2.1. Requerimientos Funcionales (RF)**

#### **2.2.2. Requerimientos No Funcionales (RNF)**

#### **2.2.3. Herramientas Específicas para RNF**

## **Módulo 3: Proceso y Técnicas de Elicitación (Obtención) de Requerimientos.**

### **3.1. Definición y Alcance de la Elicitación**

### **3.2. El Proceso de Elicitación: Actividades y Ciclo.**

### **3.3. Técnicas Clave de Elicitación.**

#### **3.3.1. Técnicas de Interacción Directa.**

#### **3.3.2. Técnicas Basadas en Modelos y Escenarios.**

#### **3.3.3. Elicitación Especializada de Requerimientos No Funcionales (RNF)**

## **Módulo 4: Análisis y Modelado de Requerimientos**

### **4.1 Propósito del Análisis y el Rol del Modelo**

## **4.2 El Proceso del Ciclo de Análisis**

## **4.3 El Modelado de Requerimientos: Tipos y Enfoques.**

### **4.3.1 Modelos Basados en Escenarios**

### **4.3.2 Modelos de Información y Clases.**

### **4.3.3 Modelos Orientados al Flujo y Comportamiento.**

## **4.4 Negociación y Priorización de Requerimientos.**

## **Módulo 5: Casos Prácticos y Ejemplos Aplicados.**

### **5.1. Sistema de Gestión de Biblioteca Universitaria**

### **5.2. Aplicación Móvil de Delivery de Comida (WebApps)**

### **5.3. Sistema de Gestión de Incidencias para Soporte Técnico (Ejemplo de Especificación Formal)**

## **Módulo 6: Gestión y Mantenimiento de Requerimientos.**

### **6.1. Control de Cambios**

### **6.2. Trazabilidad**

### **6.3. Herramientas y Tecnologías**

## **Módulo 7: Validación y Verificación de Requerimientos**

### **7.1. Distinción entre Verificación y Validación (V&V)**

### **7.2. Actividades y Comprobaciones de la Validación de Requerimientos.**

### **7.3. Técnicas de Validación de Requerimientos**

#### **7.3.1. Revisiones de Requerimientos e Inspecciones**

#### **7.3.2. Prototipado Operacional**

#### **7.3.3. Pruebas de Aceptación (Generación de Casos de Prueba)**

#### **7.3.4. Validación del Modelo (Pruebas Conceptuales)**

### **7.4. Verificación y Validación de Atributos de Calidad (RNF)**

## **BIBLIOGRAFÍA**

## **Fundamento para los requerimientos del software. Técnicas para obtener requerimientos**

### **Módulo 1: Introducción a los requerimientos del software.**

**Objetivo de aprendizaje:** El objetivo general de este tutorial es proporcionar a los estudiantes de la Facultad de Ingeniería de la UNAM un conocimiento integral y práctico sobre la Ingeniería de Requerimientos (IR), centrándose en las técnicas clave para la obtención de requisitos.<sup>[6]</sup>

Este recurso busca establecer la disciplina de la IR como el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable y especificarla sin ambigüedades, aspectos cruciales para el desarrollo exitoso de productos de software.<sup>[11]</sup>

Al completar el estudio de este material, los alumnos estarán capacitados para lograr los siguientes objetivos específicos:

1. Comprender la Importancia Crítica y Fundamentos de los Requerimientos de Software.

- Entender la disciplina de IR: Se definirá la Ingeniería de Requerimientos (IR) como la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo de software. Es una actividad de intervención humana, en la que participan las partes interesadas mediante un proceso orientado a solicitar, comprender y modelar las necesidades del cliente. En definitiva, la IR es el conjunto de actividades para descubrir, documentar y mantener un conjunto de requisitos.<sup>[1] [5] [6] [9]</sup>

- Reconocer el impacto del fracaso: Se resaltarán que la IR es la fase más crítica y compleja en el desarrollo de sistemas, y es la fase más propensa a errores. Un error en la etapa de

requisitos puede ser de 10 hasta 100 veces más costoso que un error en el código. Si cuesta \$1 localizar un error durante la IR, puede costar \$10 durante la programación y hasta \$200 después de la entrega del sistema.<sup>[12]</sup>

- Diferenciar los tipos de requerimientos (RF y RNF):
  - Se distinguirán los Requerimientos Funcionales (RF), que son enunciados acerca de los servicios que el sistema debe proveer. Describen el comportamiento y los datos que el sistema administrará.<sup>[8]</sup>
  - Los Requerimientos No Funcionales (RNF) son limitaciones o restricciones sobre los servicios o funciones que ofrece el sistema. Se relacionan con propiedades emergentes del sistema como fiabilidad, tiempo de respuesta y uso de almacenamiento.<sup>[10]</sup>
  - Se abordará la discrepancia de consenso sobre el significado de los RNF, entendiendo que son considerados como requerimientos de calidad y restricciones. Se debe notar la importancia de los RNF ya que, si no se cumplen, harían que todo el sistema sea inútil.<sup>[10]</sup>
  - Se enseñará que los RNF pueden clasificarse en: de producto (comportamiento), de organización (políticas existentes) y externos (legislación).<sup>[6][10]</sup>

## 2. Administrar y Aplicar Eficazmente las Técnicas de Obtención (Elicitación).

- Desarrollar un enfoque proactivo de obtención: Los estudiantes comprenderán que el término "ingeniería de requerimientos" es más adecuado que "relevamiento". Esto implica adoptar una actitud proactiva para identificar todos los problemas y obtener los requerimientos de manera exhaustiva y precisa, en lugar de esperar a que aparezcan fácilmente.<sup>[7]</sup>
- Dominar el proceso y las técnicas de elicitación:
  - La elicitación, también llamada Descubrimiento de Requerimientos, es el proceso de interactuar con los participantes para descubrir sus requerimientos.<sup>[7]</sup>
  - Se presentará el proceso iterativo de elicitación y se abordarán las técnicas más comunes: Entrevistas (conversaciones cara a cara), Cuestionarios o Encuestas (útiles para un gran número de *stakeholders*), Observación (para descubrir requerimientos no expresados), Análisis de Documentación Existente, Talleres de Requisitos (sesiones cortas e intensivas para consenso) y Prototipos de Exploración (versiones parciales del software para validar requisitos).<sup>[12][6]</sup>
- Aplicación a Requerimientos No Funcionales (RNF): Se abordará la dificultad de elicitar RNF, ya que las metodologías de desarrollo Web carecen de técnicas específicas. Se introducirá el uso de Plantillas de Elicitación de RNF de Calidad y Restricciones como herramientas que brindan un conjunto de preguntas para adquirir conocimiento relevante del dominio del problema.<sup>[3]</sup>

## 3. Modelar y Gestionar Requerimientos para la Calidad del Software.

- Modelado y documentación: Se enseñará que el modelado de requerimientos tiene como objetivo describir los dominios de información, función y comportamiento que se requieren para un sistema. El producto final de esta fase es el Documento de Especificación de Requerimientos de Software (ERS), que es un comunicado oficial de lo que deben implementar los desarrolladores. Este documento es la base para el diseño, codificación y pruebas del sistema.<sup>[6]</sup>
- Validación y Verificación (V&V): Se diferenciará la Verificación (punto de vista del desarrollador, asegurando que el software satisfaga los requisitos especificados) de la Validación (punto de vista del cliente, asegurando que se cumplan sus necesidades). Se enfatizará que la validación se logra mediante Revisiones de requerimientos, Creación de prototipos y generación de casos de prueba.<sup>[10]</sup>

- **Gestión del cambio (Volatilidad y Trazabilidad):** Se abordará que los requisitos tienen la característica de volatilidad, lo que dificulta su seguimiento. Por lo tanto, se requiere la Gestión de Requerimientos, que es el conjunto de actividades que ayudan a identificar, controlar y dar seguimiento a los requerimientos y a sus cambios. Esto incluye la Trazabilidad, que documenta la vida de cada requisito y permite vincular su origen con los elementos de diseño y código fuente para apoyar el análisis del impacto de los cambios.<sup>[1][6]</sup>

**Planteamiento del problema:** El desarrollo de software aún enfrenta una alta tasa de fracasos y presenta falencias en calidad, eficiencia y seguridad. La principal causa de estos problemas radica en la mala o insuficiente comprensión de los requerimientos.<sup>[1][11]</sup>

La Ingeniería de Requerimientos (IR) es considerada la fase más crítica y compleja del proceso de *software*, y un fallo en esta etapa afecta el resto del ciclo de vida.<sup>[1][6]</sup>

#### 1. Consecuencias y Riesgos de una Deficiente Elicitación

La falta de una IR sistemática y proactiva conlleva riesgos significativos:

- **Alto Costo de Corrección:** Los errores cometidos en la etapa inicial de requisitos son los más costosos de corregir en fases posteriores. Se ha calculado que si cuesta \$1 subsanar un error durante la IR, puede costar \$10 durante la programación y hasta \$200 después de la entrega del sistema.<sup>[12]</sup>
- **Volatilidad y Ambigüedad:** Los requerimientos son a menudo ambiguos, contradictorios o incompletos. Además, tienen la característica de volatilidad, lo que dificulta su seguimiento y genera la necesidad de una Gestión del Cambio constante.<sup>[1]</sup>
- **Responsabilidad del Profesional:** Aunque los usuarios tienen conocimiento de su trabajo, no siempre están capacitados para elaborar especificaciones detalladas. La responsabilidad principal recae, por lo tanto, en el ingeniero de software para aplicar buenas prácticas y construir requisitos claros y precisos.<sup>[7]</sup>

#### 2. El Desafío de la Elicitación y las Técnicas

La obtención de requerimientos es un desafío complejo. Los requerimientos no siempre son fáciles de obtener, pues los usuarios pueden no tener claras sus necesidades, no saber expresarlas correctamente, u omitir información que consideran "obvia".

El reto es mayor con los Requerimientos No Funcionales (RNF), que definen la calidad y las restricciones del sistema. La comunidad de IR no tiene consenso sobre su definición o clasificación, y la obtención tardía de RNF puede llevar a grandes, complejos y costosos cambios, haciendo que todo el sistema sea inútil si no se cumplen.<sup>[10]</sup>

### Módulo 2: Fundamentos de los Requerimientos.

El requerimiento es una noción, característica o descripción de algo que el usuario concibe para ser ejecutado por un sistema con el objetivo de optimizar su proceso de negocio. En un contexto más formal, un requisito es una declaración que identifica un atributo, una capacidad o una característica del sistema que tiene valor y utilidad para el cliente, la organización o el usuario.<sup>[1][8]</sup>

Los requerimientos actúan como pautas para que los desarrolladores creen un producto funcional que satisfaga las necesidades de los usuarios. También son descripciones de los servicios que un sistema *software* debe proporcionar y de las limitaciones con las que debe operar.<sup>[1]</sup>

#### 2.1. Propósitos y Beneficios Clave

El proceso de Ingeniería de Requerimientos (IR) y, en particular, la correcta definición de los requisitos, son cruciales para el éxito del proyecto. La correcta obtención de los requerimientos del sistema es uno de los aspectos clave en la construcción de proyectos de *software*. Si los cimientos de un edificio son débiles, todo el edificio fallará; lo mismo sucede con el *software* y sus requerimientos.<sup>[6]</sup>

Los principales propósitos y beneficios de una sólida definición de requerimientos son:

- **Asegurar el Éxito del Producto:** Los requisitos bien entendidos son la base para determinar el éxito del software implementado.<sup>[6]</sup>
- **Alineación y Visión Común:** Permiten que todas las partes interesadas (*stakeholders*) -clientes, usuarios, desarrolladores y gerentes de proyecto- comprendan y compartan una visión común del sistema. Es necesaria una representación y comprensión clara y consensuada de ellos.<sup>[1]</sup>
- **Guía para el Desarrollo:** Los requerimientos son la referencia principal para las etapas de diseño, implementación, prueba y mantenimiento. Son la base para la elección del lenguaje de programación y de las herramientas para las demás fases del ciclo de vida.<sup>[1]</sup>
- **Criterios de Validación:** Sirven como criterio de evaluación para verificar si el producto final satisface las necesidades planteadas. Además, un requisito es inequívoco, comprobable, medible y necesario para la aceptabilidad del producto final.<sup>[1]</sup>
- **Control del Alcance (Scope Creep):** Ayudan a definir los límites del sistema, evitando agregar funcionalidades no planificadas.<sup>[9]</sup>
- **Gestión de Riesgos y Costos:** Una definición clara de requerimientos reduce la posibilidad de fallos técnicos o malentendidos con el cliente. Los errores cometidos en esta etapa inicial suelen ser los más costosos de corregir en fases posteriores.

## 2.2. Clasificación de Requerimientos.

De forma general, los requisitos se clasifican en funcionales y no funcionales. Esta clasificación es la más frecuente en la literatura de Ingeniería de Software.<sup>[1] [6]</sup>

En el contexto de los Requisitos de la Solución (aquellos que describen las características de la solución), se distinguen claramente los Requisitos Funcionales y No Funcionales, especialmente cuando los requisitos describen una solución de desarrollo de *software*.<sup>[8]</sup>

### 2.2.1. Requerimientos Funcionales (RF).

Los Requerimientos Funcionales (RF) describen el comportamiento observable del sistema bajo diversas condiciones.<sup>[5]</sup>

- **Definición:** Los RF son enunciados acerca de los servicios que el sistema debe proveer. Describen el comportamiento y los datos que el sistema administrará. También definen las funciones que el sistema y sus componentes deben ejecutar.<sup>[1] [6] [8]</sup>
- **Interacción:** Describen una interacción entre el sistema y su ambiente. Describen cómo debe comportarse el sistema ante un estímulo.<sup>[12]</sup>
- **Forma de Expresión:** Se expresan en la forma general de: "el sistema deberá hacer (algún requisito)".<sup>[1]</sup>
- **Tipos de RF:** Los requerimientos funcionales se pueden dividir en:
  - **Requerimientos de Usuario:** Son declaraciones en lenguaje natural y diagramas, de los servicios que se espera que el sistema provea y de las restricciones bajo las cuales debe operar.<sup>[6]</sup>

- Requerimientos del Sistema: Establecen con detalle los servicios y restricciones del sistema. El documento que los contiene (especificación funcional) tiene que definir con exactitud lo que se implementará.<sup>[6][10]</sup>
- Ejemplos de RF:
  - El sistema debe permitir el registro de todos los clientes de la tienda.
  - El sistema debe permitir la generación de la boleta de venta y factura.
  - El módulo deberá generar el tipo de planilla y la selección de empleados.
  - A cada pedido se le deberá asignar un identificador único (ID\_PEDIDO).<sup>[8]</sup>
  - El usuario debe poder iniciar sesión con su correo y contraseña.

### 2.2.2. Requerimientos No Funcionales (RNF).

Los Requerimientos No Funcionales (RNF) especifican cómo de bien o cómo lo hace el producto.

- Definición y Naturaleza: Los RNF son una restricción sobre el sistema o su proceso de producción. Describen condiciones ambientales bajo las cuales la solución debe permanecer activa por determinados períodos de tiempo o cualidades que los sistemas deben tener a nivel operacional.<sup>[8]</sup>
- Propiedades Emergentes: Se relacionan con las propiedades emergentes del sistema, como la fiabilidad, el tiempo de respuesta y el uso de almacenamiento.<sup>[8] [10]</sup>
- Significancia Crítica: Los RNF a menudo son más significativos que los requerimientos funcionales individuales. El fracaso en cumplir con un RNF importante (como la fiabilidad) haría que todo el sistema fuera inútil.<sup>[10]</sup>
- Falta de Consenso: La noción de RNF no es homogénea, y hay falta de consenso en cuanto a su definición y clasificación en la literatura de Ingeniería de Requerimientos. Por ello, se pueden establecer las bases conceptuales para considerarlos como requerimientos de calidad y restricciones.<sup>[3]</sup>
  - RNF como Calidad: Representan cualidades que el sistema debe tener, tales como precisión, usabilidad, seguridad, rendimiento, confiabilidad (*performance*), entre otras. El estándar de calidad internacional ISO/IEC 9126 incluye características consideradas como RNF de calidad.<sup>[3][11]</sup>
  - RNF como Restricciones: Son las restricciones globales sobre el sistema, un requerimiento funcional, el proceso de desarrollo o sobre el proceso de despliegue.<sup>[3]</sup>
- Forma de Expresión: Se expresan en la forma general de “el sistema será (algún requisito)”.<sup>[1]</sup>
- Clasificación de Sommerville: Los RNF se clasifican en:
  - Requerimientos de Producto: Especifican o restringen el comportamiento del *software*. Incluyen rendimiento, fiabilidad, seguridad y usabilidad.<sup>[6][10]</sup>
  - Requerimientos de Organización: Se derivan de las políticas y procedimientos existentes en la organización del cliente y del desarrollador. Incluyen requerimientos operacionales y de desarrollo.<sup>[6] [10]</sup>
  - Requerimientos Externos: Derivan de factores externos al sistema y su proceso de desarrollo. Incluyen requerimientos regulatorios y legislativos.<sup>[6] [10]</sup>
- Ejemplos de RNF:
  - Rendimiento: La búsqueda de libros debe devolver resultados en menos de 3 segundos.
  - Seguridad: Todas las contraseñas de usuario deben almacenarse cifradas usando el algoritmo SHA-256.
  - Disponibilidad: El sistema deberá soportar una disponibilidad de sus servidores de 24x7 al 98% de disponibilidad.



- Portabilidad: La aplicación debe funcionar en los navegadores Chrome, Firefox.

### 2.2.3. Herramientas Específicas para RNF

El tratamiento de los RNF se dificulta debido a la falta de lineamientos, técnicas y/o herramientas específicas de soporte en los enfoques utilizados para el desarrollo de aplicaciones Web.<sup>[3]</sup>

- Plantillas de Elicitación: Para la captura y especificación de RNF Web (de calidad y restricciones) se utilizan plantillas diseñadas específicamente para ello. Estas plantillas proveen un marco de referencia y una guía para la búsqueda y adquisición del conocimiento relevante del dominio del problema relacionado a los RNF.<sup>[3]</sup>
- Clasificación en Plantillas: La plantilla para la elicitación de RNF de Calidad contiene varias características y subcaracterísticas del estándar ISO/IEC 9126-1. Las características incluidas son Funcionalidad (Precisión, Interoperabilidad, Seguridad), Confiabilidad, Usabilidad, Eficiencia y Portabilidad.<sup>[3]</sup>

## Módulo 3: Proceso y Técnicas de Elicitación (Obtención) de Requerimientos.

### 3.1. Definición y Alcance de la Elicitación

La **Elicitación de Requerimientos** es la primera fase del proceso de desarrollo de requerimientos. También se la conoce como **Descubrimiento de Requerimientos**.<sup>[6] [7] [10]</sup>

Esta actividad es la más crucial y desafiante en el desarrollo de *software*, ya que sienta la **base para la definición del alcance** de las mejoras a efectuar dentro de una organización.<sup>[6] [8]</sup>

#### Definición y Enfoque Proactivo

1. **Objetivo de la Elicitación:** En esta fase, el equipo de desarrollo y los *stakeholders* (interesados) trabajan juntos para **identificar, articular y entender los requerimientos** de la aplicación. Implica comprender el **dominio de la aplicación**, los servicios que se deben proveer, los problemas que se buscan resolver y las necesidades de los usuarios.<sup>[6] [10]</sup>

2. **Actitud del Ingeniero:** El término "Ingeniería de Requerimientos" es más apropiado que "relevamiento". El ingeniero de requerimientos debe abandonar una actitud pasiva de sólo registrar lo que se comunica y adoptar una **postura proactiva**, buscando activamente los requisitos en lugar de esperar a que estos aparezcan fácilmente.<sup>[7] [10]</sup>

3. **Proceso Colaborativo:** La elicitación combina elementos de **solución de problemas, elaboración, negociación y especificación**. A fin de estimular un enfoque colaborativo, los participantes trabajan juntos para identificar el problema y especificar un conjunto preliminar de requerimientos.<sup>[11]</sup>

### 3.2. El Proceso de Elicitación: Actividades y Ciclo

La elicitación y el análisis de requerimientos es un **proceso iterativo** que se representa mejor como una **espiral de actividades**. Los ingenieros deben elegir las técnicas de elicitación que mejor se ajusten al contexto del problema y a las características de las partes interesadas.<sup>[6]</sup>

#### Fases del Proceso Iterativo

Para obtener los requerimientos de forma efectiva, se sugiere un ciclo que se repite para profundizar el entendimiento:<sup>[6]</sup>

1. **Elegir y planificar** las técnicas de elicitación de requerimientos.<sup>[6]</sup>
2. **Establecer metas, expectativas** y prepararse (preparación de observación, cuestionarios, etc.).<sup>[6]</sup>
3. **Elicitar** los requerimientos (aplicación de la técnica).<sup>[6]</sup>
4. **Verificar y corregir** los hallazgos.<sup>[6]</sup>
5. **Repetir** los pasos anteriores para profundizar el entendimiento.<sup>[6]</sup>

#### Actividades Clave de Adquisición y Análisis

La actividad de Adquisición y Análisis de requerimientos sigue un proceso iterativo que típicamente incluye:

1. **Descubrimiento de Requerimientos:** Proceso de **interactuar con los participantes** del sistema para descubrir sus requerimientos.<sup>[7] [10]</sup>
2. **Clasificación y Organización:** Tomar la compilación no estructurada de requerimientos, **agruparlos** y organizarlos en grupos coherentes.<sup>[7] [10]</sup>
3. **Priorización y Negociación:** Asignar un precedente que ordene o clasifique un requerimiento sobre otro.<sup>[10]</sup>
4. **Especificación de Requerimientos:** Documentar los requerimientos e ingresarlos para la siguiente ronda de la espiral.<sup>[10]</sup>

### 3.3. Técnicas Clave de Elicitación.

La elicitación utiliza numerosas **técnicas complementarias** que pueden ser de interacción directa o basadas en el uso de modelos.<sup>[7] [8]</sup>

#### 3.3.1. Técnicas de Interacción Directa.

Estas técnicas requieren la participación activa y el contacto con los interesados.

**Entrevistas:** Consiste en realizar conversaciones cara a cara para obtener información específica. Las entrevistas y los cuestionarios deben prepararse.<sup>[3] [7] [8]</sup>

**Observación:** El enfoque puede ser activo, donde el ingeniero se involucra en las tareas y forma parte del equipo de trabajo. Se debe determinar las personas a observar y elaborar cuestionarios que se aplicarán durante o después de las actividades. Es útil para observar el entorno donde se instalará el sistema.<sup>[6] [12]</sup>

**Talleres de Requisitos (Workshops):** Son reuniones intensivas con partes interesadas seleccionadas y expertos para definir, refinar y obtener consenso.<sup>[8]</sup>

**Análisis de Documentos:** Consiste en obtener requerimientos mediante el estudio de la documentación existente.<sup>[8] [3]</sup>

**Encuestas / Cuestionarios:** Se utilizan para el descubrimiento de requerimientos y son listadas como una técnica.<sup>[8] [7]</sup>

### 3.3.2. Técnicas Basadas en Modelos y Escenarios.

**Prototipos (Prototipado):** Un prototipo permite obtener **retroalimentación** de los requisitos, especialmente en cuanto a las interfaces. El prototipo inicial se retroalimenta y genera nuevas versiones hasta que el usuario aprueba. El prototipo funcional muestra **aspectos imprevistos** del comportamiento del sistema, lo que ayuda a cerrar la especificación. Los **prototipos desechables** (*throw-away prototypes*) son recomendados como la **forma de interacción casi excluyente** entre los ingenieros y el cliente durante la obtención de requisitos funcionales.<sup>[12] [8]</sup>

**Escenarios y Casos de Uso:** Los requerimientos pueden expresarse como **escenarios (historias de usuario)**, especialmente en metodologías ágiles como la Programación Extrema (XP). Un caso de uso es un escenario específico desde el punto de vista de un actor, y describe la interacción entre el sistema y su ambiente. Son una técnica útil para el descubrimiento de requerimientos.<sup>[10] [11] [12] [7]</sup>

**Despliegue de la Función de Calidad (DFC):** Técnica que traduce las necesidades del cliente en requerimientos técnicos. Se centra en maximizar la satisfacción del cliente y enfatiza en entender lo que resulta valioso para él, desplegando esos valores a través del proceso de ingeniería.<sup>[11]</sup>

### 3.3.3. Elicitación Especializada de Requerimientos No Funcionales (RNF)

La obtención de RNF es particularmente desafiante, ya que estos requerimientos son a menudo **confusos, ambiguos y más difíciles de prototipar** que los funcionales.<sup>[12]</sup>

#### 1. Uso de Checklists y Plantillas:

- La complejidad de los RNF en aplicaciones *web* ha llevado a la falta de lineamientos y herramientas específicas.<sup>[3]</sup>

- Se utilizan **Plantillas de Elicitación de RNF de Calidad y Restricciones** para guiar al ingeniero y adquirir el conocimiento relevante del dominio del problema.<sup>[3]</sup>

- La **Plantilla de RNF de Calidad** se basa en características y subcaracterísticas del estándar internacional **ISO/IEC 9126-1**. Las características incluidas son Funcionalidad (Precisión, Interoperabilidad, Seguridad), Confiabilidad, Usabilidad, Eficiencia y Portabilidad.<sup>[3]</sup>

- Estas preguntas ayudan a identificar RNF como la Precisión, Seguridad, Tolerancia a Fallas y Capacidad de instalación.<sup>[3]</sup>

#### 2. Escenarios de Atributos de Calidad:

◦ Debido a que es complejo o costoso prototipar atributos de calidad (como disponibilidad o seguridad), se utilizan escenarios de atributos de calidad como una técnica efectiva.<sup>[12]</sup>

◦ Un **Escenario de Atributo de Calidad** es un requerimiento no funcional específico sobre ese atributo de calidad. Se compone de seis elementos:

- **Estímulo:** La condición que llega al sistema.<sup>[12]</sup>
- **Entorno:** Las condiciones bajo las cuales se da el estímulo.<sup>[12]</sup>
- **Artefacto:** El elemento del sistema afectado (puede ser todo el sistema).<sup>[12]</sup>
- **Respuesta:** La actividad que se lleva a cabo.<sup>[12]</sup>
- **Medida de la Respuesta:** Debe ser **medible** de alguna forma para que el requerimiento pueda ser testeado.<sup>[12]</sup>
- **Origen:** El actor que originó el requerimiento (no citado en este fragmento, pero es un elemento del Patrón de RNF según la Matriz).<sup>[3]</sup>

◦ Una vez que el cliente manifiesta interés en un atributo, el ingeniero debe **derivar escenarios concretos** a partir de escenarios generales para obtener los RNF específicos del cliente.<sup>[12]</sup>

## Módulo 4: Análisis y Modelado de Requerimientos

El **Análisis de Requerimientos** es la fase posterior a la Elicitación y es fundamental para construir sobre los requerimientos básicos que se establecieron durante la concepción, la indagación y la negociación. Su principal meta es refinar los requisitos, asegurando que todos los *stakeholders* (interesados) los comprendan y examinen las posibles deficiencias, como errores u omisiones.<sup>[6] [11]</sup>

### 4.1 Propósito del Análisis y el Rol del Modelo

El Análisis incluye la descomposición detallada de los requisitos de alto nivel, la evaluación de la viabilidad y la negociación de prioridades. El objetivo final es desarrollar requisitos con suficiente calidad y detalle para que el personal técnico pueda proceder con el diseño, la construcción y las pruebas de manera realista.<sup>[11] [6]</sup>

1. **El Modelo como Puente:** El modelo de análisis sirve como un **puente** entre la descripción general del sistema (que se centra en el negocio) y el diseño de *software*.<sup>[11]</sup>

2. **Objetivos Clave:** El modelo de requerimientos debe lograr tres objetivos principales: describir lo que requiere el cliente, establecer la base para la creación de un diseño de *software* y definir un conjunto de requerimientos que puedan **validarse** una vez construido el *software*.<sup>[11]</sup>

3. **Abstracción Necesaria:** En esta fase, el nivel de abstracción debe ser elevado, centrándose en los requerimientos que son **visibles dentro del dominio del problema** o

**negocio.** Se deben retrasar las consideraciones de la infraestructura o los modelos no funcionales hasta la etapa de diseño.<sup>[11]</sup>

## 4.2 El Proceso del Ciclo de Análisis

La Ingeniería de Requerimientos se desarrolla mediante una **elaboración progresiva o incremental**. El análisis se integra continuamente con la elicitación y la especificación.

Las actividades clave del ciclo de análisis y modelado de requerimientos son:

1. **Modelado del Negocio:** Determinar si los modelos de negocios son necesarios y, en caso afirmativo, crearlos y verificar su exactitud a medida que evolucionan.<sup>[6]</sup>
2. **Definir el Alcance del Proyecto:** Se delimita el alcance del sistema. Es crucial crear modelos que describan el alcance y revisar que no haya defectos o inconsistencias entre los requerimientos.<sup>[6]</sup>
3. **Crear Modelos Detallados:** Desarrollar los modelos de requerimientos del usuario al detalle, incluyendo elementos basados en escenarios, clases, flujo y comportamiento.<sup>[6]</sup>
4. **Negociar Prioridades:** Analizar y priorizar los requerimientos con los *stakeholders* para tomar decisiones sobre su importancia.<sup>[6]</sup>

## 4.3 El Modelado de Requerimientos: Tipos y Enfoques.

El resultado del análisis se plasma en los modelos de requerimientos, que son las necesidades de los usuarios representadas por diagramas, texto estructurado o una combinación de ambos. Dado que cualquier "vista" de los requerimientos es insuficiente para comprender un sistema complejo, se utiliza una combinación de representaciones.<sup>[11]</sup>

Los principales tipos de modelos que se crean en esta fase son los siguientes.

### 4.3.1 Modelos Basados en Escenarios

Estos modelos ilustran cómo el usuario interactúa con el sistema.

- **Casos de Uso:** Describen un **escenario específico** desde el punto de vista de un actor definido. Los casos de uso son una técnica útil para el descubrimiento de requerimientos. Describen la secuencia específica de actividades que ocurren cuando se utiliza el *software* y deben ser lo suficientemente detallados para servir de base para el diseño.<sup>[11][9]</sup>
- **Historias de Usuario:** Al igual que los casos de uso, se utilizan para expresar requerimientos como escenarios.<sup>[10]</sup>

### 4.3.2 Modelos de Información y Clases.

Estos modelos se centran en el dominio de la información y los objetos de datos que el sistema debe manipular.

- **Modelo de Datos / Modelo Entidad-Relación (MER):** Se crea si los requerimientos implican la necesidad de hacer interfaz o manipular una base de datos. El MER describe las

necesidades de información del proceso de negocio, detallando **entidades**, **atributos** (características de la entidad) y **relaciones** entre ellas. El modelo de datos es un elemento crucial del modelo de análisis.<sup>[11]</sup>

- **Modelos Orientados a Clases:** Representan las clases orientadas a objetos, sus atributos, sus operaciones y la manera en que estas clases colaboran para cumplir con los requerimientos.<sup>[11]</sup>

#### 4.3.3 Modelos Orientados al Flujo y Comportamiento.

Estos modelos describen la dinámica del sistema en respuesta a eventos y cómo se transforma la información.

- **Modelos Orientados al Flujo:** Representan los elementos funcionales del sistema y la forma en que transforman los datos a medida que estos fluyen a través del sistema. Esta técnica utiliza el Diagrama de Flujo de Datos (DFD) para ilustrar cómo se transforma la entrada en salida.<sup>[11]</sup>
- **Modelos de Comportamiento:** Ilustran la manera en que el *software* se comporta como consecuencia de eventos externos. El modelo de comportamiento ilustra los estados del sistema y el efecto que los eventos tienen sobre dichos estados.<sup>[11]</sup>

#### 4.4 Negociación y Priorización de Requerimientos.

La negociación es una tarea fundamental que ocurre en paralelo con el análisis y la elaboración.

- **Necesidad de Negociación:** La IR es un proceso complejo porque los requisitos son a menudo confusos, no verificables ni validables. La negociación busca resolver conflictos y ambigüedades. La meta de esta etapa es definir una **solución razonable**.<sup>[11]</sup>
- **Priorización:** No todos los requisitos son igual de importantes, y rara vez hay suficiente tiempo y dinero para desarrollar todos. La **priorización de requerimientos** es el proceso de asignar un precedente que ordene o clasifique un requerimiento sobre otro.<sup>[6]</sup>
- **Toma de Decisiones:** Los *stakeholders* necesitan la priorización para tomar **decisiones inteligentes y defendibles** sobre cuáles requisitos implementar o aplazar. La priorización se puede realizar utilizando criterios como el valor de negocio y la importancia para el cliente.<sup>[6]</sup>

Una vez que los requerimientos han sido analizados, modelados y negociados, están listos para pasar a la siguiente fase: la **Especificación** (documentación formal).

#### Módulo 5: Casos Prácticos y Ejemplos Aplicados.

El estudio de casos prácticos y ejemplos es esencial para la Ingeniería de Requerimientos (IR), ya que permite al estudiante pasar de la teoría a la aplicación real de las técnicas de obtención, análisis y modelado. Al analizar escenarios específicos, se puede determinar qué requerimientos (funcionales y no funcionales) son prioritarios y qué modelos de análisis son más adecuados para la solución.<sup>[7] [6]</sup>

A continuación, se analizan los desafíos y requerimientos esenciales que se esperarían en diferentes tipos de sistemas:

### 5.1. Sistema de Gestión de Biblioteca Universitaria

Los sistemas de gestión son un tipo de aplicación que requiere un diseño bastante detallado antes de comenzar el desarrollo, debido a la posible vinculación con normativas o reglas complejas. La IR en estos proyectos se enfoca en establecer los elementos necesarios para planificar las actividades y garantizar que el sistema dé un verdadero soporte y agregue valor al proceso de la organización.<sup>[7] [6]</sup>

En un Sistema de Gestión de Biblioteca, se aplicarían los siguientes conceptos:

- **Modelado Orientado al Flujo y Comportamiento:** Se utilizaría el Diagrama de Flujo de Datos (DFD) para ilustrar cómo se transforma la entrada (ej. solicitud de préstamo) en una salida (ej. libro entregado y registro actualizado) [368, 7.2.1]. El DFD muestra los elementos funcionales del sistema y cómo transforman los datos que fluyen a través de él.<sup>[11]</sup>
- **Modelado Orientado a Clases:** Se definen las clases de análisis del dominio, como *Libro*, *Usuario* o *Préstamo*, que serán manipuladas por el sistema. La salida de la fase de diseño define las estructuras de datos y las arquitecturas.<sup>[7]</sup>
- **Requerimientos de la Organización (RNF):** Un ejemplo de RNF organizacional es aquel que deriva de las políticas y procedimientos existentes en la organización. En este caso, podrían ser las reglas específicas de la universidad sobre el tiempo máximo de préstamo o el número de libros que un estudiante puede tener. Por ejemplo, el requerimiento de que el sistema funcione de acuerdo con un horario laboral específico (18:00 a 19:00 horas) se clasificaría como un Requerimiento de la Organización.<sup>[8]</sup>

### 5.2. Aplicación Móvil de Delivery de Comida (WebApps)

El desarrollo de aplicaciones Web (*Webapps*) y móviles presenta desafíos específicos, ya que el enfoque se desplaza fuertemente hacia la **experiencia del usuario** y los **Requerimientos No Funcionales (RNF)**. La complejidad de los requerimientos para aplicaciones Web demanda herramientas y técnicas acordes a su naturaleza.

- **Prioridad de RNF (Atributos de Calidad):** Los RNF se relacionan con propiedades emergentes, y el fracaso en cumplir con un RNF importante podría hacer que **todo el sistema sea inútil**.<sup>[10] [3]</sup>
- **Ejemplos de RNF de Calidad y Restricción para WebApps:** El caso de una aplicación de *delivery* ilustra la importancia de los RNF en este dominio:
  - **Rendimiento:** La aplicación (*app*) debe cargar el menú en menos de 2 segundos. Esta métrica debe ser verificable, ya que la **Rapidez** se mide en *Transacciones por segundo procesadas* o *Tiempo de respuesta usuario/evento*.<sup>[10]</sup>
  - **Seguridad:** Los datos de pago deben transmitirse usando TLS 1.3.

- **Usabilidad:** Un usuario nuevo debe poder hacer un pedido en menos de 5 minutos. La **Facilidad de uso** se puede medir por el *Tiempo de capacitación*.<sup>[10]</sup>

- **Portabilidad:** La *app* debe funcionar en Android (versión 10+) e iOS (versión 14+).

- **Técnicas de Validación:** Para asegurar la calidad de la interfaz y la funcionalidad, se usa el **prototipado interactivo** (ej. usando herramientas como *Marvel App* para simular el flujo de pedido). Un prototipo ayuda a validar los requisitos y a que los usuarios obtengan nuevas ideas para requerimientos.<sup>[10]</sup>

### 5.3. Sistema de Gestión de Incidencias para Soporte Técnico (Ejemplo de Especificación Formal)

El éxito en la IR culmina en la **Especificación de Requerimientos**, un proceso de elaboración, refinamiento y organización de los requisitos en un documento formal. Utilizaremos el ejemplo de un sistema de gestión de trámites (similar a la gestión de incidencias) para ilustrar la formalidad requerida.<sup>[6]</sup>

- **El Documento de Especificación (ERS):** El producto final del proceso de requisitos es un Documento de Especificación de Requerimientos de Software (ERS). Este documento es un **comunicado oficial** de lo que deben implementar los desarrolladores.<sup>[6] [10]</sup>

- **Estructura Formal:** Las plantillas para la especificación se diseñan siguiendo estándares como el **IEEE 830-1998**. El ERS debe ser la base para la planificación del proyecto, el diseño, la codificación, las pruebas y la documentación de usuario.<sup>[10]</sup>

- **Especificación de los Requerimientos Funcionales:** Se recomienda escribir frases cortas y concisas usando imperativos para describir los RF. La estructura debe incluir la **restricción**, el **asunto** (quién o qué realiza la tarea), la **acción del verbo** y el **resultado observable**.<sup>[6]</sup>

- **Integración de la Trazabilidad:** Los requerimientos funcionales deben tener referencias a los atributos de calidad (RNF) relacionados. La **trazabilidad** es la documentación de la vida de cada requisito, lo que permite el seguimiento a su origen (necesidades) y hacia adelante (diseño, código y pruebas). La trazabilidad se utiliza para validar el diseño de la arquitectura y la consideración de los requisitos funcionales y no funcionales.<sup>[6] [8]</sup>

- **Plantillas para RNF:** La especificación de RNF Web se apoya en plantillas diseñadas a partir de bases conceptuales. Además de los RNF de Calidad, se utilizan plantillas para las **restricciones**.<sup>[3]</sup>

- **Prototipo y Validación:** La elaboración de prototipos es una técnica clave de **validación** de requisitos que involucra el diseño de pantallas en una herramienta gráfica o el maquetado. La creación de prototipos ayuda a **probar las opciones de diseño** y a **entender mejor el problema y su solución**.<sup>[8]</sup>

### Módulo 6: Gestión y Mantenimiento de Requerimientos.

La **Gestión de Requerimientos (RM)** es el conjunto de actividades que se realizan para garantizar que el producto que se está construyendo es lo que el cliente realmente necesita



y desea. Debido a que los requisitos son intrínsecamente volátiles, la gestión es una actividad continua y esencial durante todo el ciclo de vida del *software*.<sup>[6]</sup>

## 6.1. Control de Cambios

Los requerimientos tienen la característica de **volatilidad**, lo que dificulta su seguimiento. Es un riesgo cerrar la etapa de relevamiento, ya que los cambios pueden ocurrir en cualquier momento, incluso provenientes del entorno o de nueva normativa.<sup>[7]</sup>

El **cambio es inevitable**. Por lo tanto, el control de cambios es una de las tareas fundamentales de la Gestión de Requerimientos (RM).<sup>[11] [6]</sup>

### Proceso y Objetivos

La Gestión de Requerimientos es el proceso de **seguimiento del estado** de los requisitos y **controlar los cambios**. El objetivo de esta actividad debe ser **adoptar y gestionar el cambio**, no evitarlo.<sup>[3] [10]</sup>

Este proceso implica una secuencia de tareas para asegurar que la evolución del sistema sea ordenada y económica:

1. **Solicitud de Cambio:** Se define cómo las personas solicitan cambios.<sup>[11]</sup>
2. **Evaluación:** Se analiza el impacto del cambio.<sup>[11]</sup>
3. **Decisión y Ejecución:** Se decide cuándo y si implementar el cambio. Las decisiones deben tomarse de manera inteligente y defendible.
4. **Mantenimiento de Registros:** Se deben conservar registros de **lo que afecta el cambio**.<sup>[11]</sup>

Un plan complementario para la **Administración de la Configuración del Software (ACS)** debe enfocarse en los riesgos para los proyectos y el calendario.<sup>[10]</sup>

## 6.2. Trazabilidad

La trazabilidad es un elemento crítico de la gestión, pues proporciona la visibilidad necesaria para manejar la volatilidad.

- **Definición:** La **trazabilidad** es la documentación de la **vida de cada requisito**.<sup>[10] [3]</sup>
- **Requisito Trazable:** Un requisito es trazable si se le puede hacer **seguimiento a su vida útil**. Esto significa que debe poder vincularse a su **origen** hacia atrás (necesidades del cliente) y hacia adelante a otros artefactos, como el **diseño, el código y las pruebas**.<sup>[3]</sup>
- **Función Clave:** La trazabilidad apoya directamente la **gestión del cambio** al permitir comprender su impacto en los requisitos y los componentes del sistema.<sup>[3]</sup>
- **Herramientas:** La versión definitiva de la **Matriz de RNF** es un documento que permite la trazabilidad de cada Requerimiento No Funcional (RNF). Las **políticas de seguimiento**

definen las relaciones entre cada requerimiento y el diseño del sistema que debe registrarse.<sup>[10]</sup>

### 6.3. Herramientas y Tecnologías

La complejidad de la Gestión de Requerimientos requiere el uso de herramientas de apoyo, ya que esta actividad implica el procesamiento de **grandes cantidades de información** acerca de los requerimientos.

- **Tipos de Herramientas:** Las herramientas disponibles varían en sofisticación:

- **Herramientas Especializadas de Administración de Requerimientos:** *Software* diseñado específicamente para esta tarea.<sup>[10]</sup>

- **Sistemas de Bases de Datos Simples o Hojas de Cálculo:** Utilizados para el procesamiento de la información.<sup>[10]</sup>

- **Uso de Matrices de RNF:** Las herramientas permiten manejar artefactos clave como la **Matriz de RNF**, la cual se elabora en **múltiples versiones** (se mencionan 3 versiones) a lo largo del proceso.<sup>[3]</sup>

- La Matriz de RNF registra las diferencias e inconsistencias entre los Requerimientos No Funcionales candidatos.<sup>[3]</sup>

- Una columna de la Matriz de RNF se utiliza para reflejar las **relaciones** entre los RNF candidatos de calidad y de restricciones.<sup>[3]</sup>

- **Gestión de la Configuración:** Es crucial utilizar una herramienta de Administración de la Configuración del Software (ACS) para el **control de cambio y de versión**. Las herramientas de ACS proporcionan soporte en este contexto.<sup>[11]</sup>

- **Impacto de la Herramienta:** Las herramientas deben ayudar al equipo a seguir la pista de un diseño en evolución. Sin embargo, la efectividad del proceso sigue dependiendo de la capacidad del analista para manejar los inconvenientes, incluso cuando la documentación no es apropiada.<sup>[6]</sup>

## Módulo 7: Validación y Verificación de Requerimientos

La Validación y Verificación (V&V) de requerimientos es una actividad de **calidad** que garantiza la solidez del trabajo realizado en las fases previas (elicitación, análisis y especificación). Es el proceso de asegurar que se construya el **producto correcto** y que se construya **bien**.<sup>[10][11] [8]</sup>

El proceso de validación está presente en **todos los modelos** de procesos de Ingeniería de Requerimientos (IR). Su objetivo es detectar y corregir cualquier requisito **innecesario e incorrecto** antes de que el diseño y el desarrollo comiencen.<sup>[6]</sup>

### 7.1. Distinción entre Verificación y Validación (V&V)

Aunque los términos suelen usarse juntos, representan enfoques distintos y complementarios.<sup>[10]</sup>

En la **validación** se enfoca en el **punto de vista del cliente** y se asegura de que las **necesidades del cliente se cumplan**.<sup>[6]</sup> Nos preguntamos: **¿Construimos el producto correcto?**<sup>[10] [11]</sup>

En la **verificación** se enfoca en el **punto de vista del equipo de desarrollo** y asegura que el *software* satisfaga los **requisitos especificados**.<sup>[10]</sup> Nos preguntamos: **¿Construimos bien el producto?**<sup>[10] [11]</sup>

La validación es un proceso más general que va **más allá de la conformidad con la especificación**. Una especificación, aunque cumpla con el estándar, puede no reflejar los deseos o las necesidades reales del cliente.<sup>[10]</sup>

**Importancia Crítica:** La validación de requerimientos es esencial porque los errores en esta etapa, si se descubren tarde (durante el desarrollo o después de la entrega), pueden conducir a **grandes costos** por tener que rehacer el trabajo. El costo de corregir un error en etapas finales es **de 10 a 100 veces** el costo de corregirlo en etapas iniciales.<sup>[10] [6]</sup>

## 7.2. Actividades y Comprobaciones de la Validación de Requerimientos.

La validación es un proceso de comprobación de que los requisitos se especificaron de acuerdo con las necesidades de los clientes. Se realiza una vez obtenida una primera versión de la documentación de especificación.<sup>[6]</sup>

Tipos de Comprobaciones:

1. **Comprobaciones de Validez:** Se verifica que los requisitos realmente definan el sistema que el cliente **en verdad quiere**. Implica identificar si se requieren funciones adicionales o diferentes.<sup>[10]</sup>

2. **Comprobaciones de Consistencia:** Se asegura que los requerimientos **no estén en conflicto** entre sí, es decir, que no haya restricciones contradictorias o descripciones diferentes de la misma función.<sup>[10]</sup>

3. **Comprobaciones de Totalidad:** Se verifica que el documento incluya requerimientos que definan **todas las funciones y restricciones** pretendidas por el usuario.<sup>[10]</sup>

4. **Comprobaciones de Realismo:** Al usar el conocimiento de la tecnología existente, se comprueba que los requerimientos **puedan implementarse** considerando el presupuesto y la fecha de entrega.<sup>[10]</sup>

5. **Verificabilidad:** Se asegura que los requerimientos se escriban de manera que sean **comprobables**, es decir, que se pueda **escribir un conjunto de pruebas** que demuestren que el sistema entregado cumpla con cada requerimiento especificado. Un requerimiento es verificable cuando puede ser cuantificado.<sup>[10]</sup>

Preguntas Clave para la Validación (Checklist):

Es útil analizar cada requerimiento en comparación con una lista de verificación:

- ¿El requerimiento **es realmente necesario** o representa una característica agregada que tal vez no sea esencial para el objetivo del sistema?<sup>[11]</sup>
- ¿Cada requerimiento está acotado y **no es ambiguo**?<sup>[11]</sup>
- ¿Tiene **atribución** cada requerimiento (fuente clara)?<sup>[11]</sup>
- ¿Hay requerimientos **en conflicto** con otros?<sup>[11]</sup>
- ¿El requerimiento **puede someterse a prueba** una vez implementado?<sup>[11]</sup>
- ¿El modelo de requerimientos **refleja de manera apropiada** la información, la función y el comportamiento del sistema que se va a construir?<sup>[11]</sup>

### 7.3. Técnicas de Validación de Requerimientos

La validación se logra a través de una serie de pruebas y análisis, utilizando varias técnicas que se pueden aplicar individualmente o en conjunto.<sup>[10]</sup>

#### 7.3.1. Revisiones de Requerimientos e Inspecciones

La revisión técnica es el mecanismo principal de validación de los requerimientos.

- **Proceso:** Los requerimientos se analizan sistemáticamente usando un equipo de revisores (ingenieros de *software*, clientes y usuarios) que verifican si hay errores e inconsistencias.<sup>[11]</sup>
- **Inspecciones:** Constituyen verificaciones de calidad de los productos generados como parte del ciclo de vida del desarrollo.<sup>[8]</sup>
- **Revisiones de Pares:** Obligan a los analistas a centrar los esfuerzos de validación en las partes que tienen **mayor riesgo y ambigüedad**. Se utilizan listas de comprobación (*checklist*) para garantizar la calidad.<sup>[6]</sup>

#### 7.3.2. Prototipado Operacional

El prototipado es una de las técnicas más importantes, especialmente para la validación de requisitos funcionales.<sup>[12]</sup>

- **Definición:** Se muestra un modelo ejecutable del sistema a los usuarios finales y clientes, permitiéndoles **experimentar** con el modelo para constatar si cubre sus necesidades reales.<sup>[10]</sup>
- **Función Clave:** Ayuda a probar las opciones de diseño y a **entender mejor el problema y su solución**. Además, la creación de prototipos es un mecanismo poderoso para reducir la **brecha de expectativas**.<sup>[12]</sup>
- **Tipos:** El **prototipo desechable** (*throw-away prototype*) es recomendado como la forma de interacción casi excluyente con el cliente durante esta fase.<sup>[12]</sup>

### 7.3.3. Pruebas de Aceptación (Generación de Casos de Prueba)

La generación de casos de prueba es una técnica que a menudo revela problemas en los requerimientos.<sup>[10]</sup>

- **Propósito:** Si una prueba es difícil o imposible de diseñar, generalmente significa que el requerimiento será difícil de implementar, por lo que debería reconsiderarse. El desarrollo de pruebas debe realizarse a partir de los requerimientos del usuario antes de escribir cualquier código.<sup>[10]</sup>

- **Pruebas Basadas en Requerimientos:** Es un enfoque sistemático donde se diseña una prueba específica para cada requerimiento, con el fin de demostrar que el sistema implementó adecuadamente sus requerimientos (pruebas de validación).<sup>[10]</sup>

- **Pruebas de Aceptación de Usuario (UAT):** Son pruebas de validación que se ejecutan en las etapas finales. El **usuario final** es quien ejecuta las pruebas de aceptación. Estas pruebas definen la **aceptación o rechazo** final del sistema por parte de los usuarios.<sup>[6]</sup>

### 7.3.4. Validación del Modelo (Pruebas Conceptuales)

Esta técnica se denomina también **pruebas conceptuales** o **análisis lógico**.

- **Mecanismo:** Utiliza simulaciones de pruebas en lugar de casos de prueba reales para pasar por múltiples modelos de análisis.<sup>[6]</sup>

- **Objetivo:** Ayuda a descubrir la **falta de información** y a corregir errores de documentación. También permite que usuarios y miembros del equipo simulen la operación del sistema para **comprobar que los requerimientos cubren situaciones típicas** del usuario.<sup>[6]</sup>

### 7.4. Verificación y Validación de Atributos de Calidad (RNF)

Los atributos de calidad del producto (RNF) también deben ser verificados y validados mediante diferentes tipos de pruebas.

- **Pruebas de Sistemas:** Son pruebas de validación que se ejecutan en un ambiente similar al de producción y revisan el correcto funcionamiento del *software* basándose en los casos de pruebas que consignan los requisitos funcionales y **no funcionales**.<sup>[8]</sup>

- **Pruebas No Funcionales:** Se centran en verificar criterios relacionados con **pruebas de estrés, concurrencia, carga, usabilidad, escalabilidad y seguridad**. Por ejemplo, las **pruebas de carga** evalúan el comportamiento del sistema frente a una sobrecarga de datos, y las **pruebas de esfuerzo** fuerzan el sistema hasta el punto de rompimiento.<sup>[8]</sup>

- **Pruebas de Seguridad:** Validan la vulnerabilidad del sistema, revisando el cumplimiento de estándares de programación, control de acceso, canales seguros, y configuración de servidores.<sup>[8]</sup>

**Conclusión:**

La **Ingeniería de Requerimientos (IR)** se establece como el **pilar fundamental de todo proyecto de desarrollo de software**. Siendo la fase más **crítica y compleja** del ciclo de vida, la IR define la base sobre la que se construye y se garantiza la calidad, la eficiencia y la seguridad del producto final.<sup>[1]</sup>

El objetivo de este proceso sistemático es transformar las necesidades a menudo ambiguas e informales de los usuarios en una **especificación precisa y consistente**. El fracaso en esta etapa puede conducir a **pérdidas económicas considerables** y el costo de corregir un error de requerimiento en fases posteriores es **mucho mayor** que reparar errores de diseño o codificación.<sup>[10][12]</sup>

Para mitigar estos riesgos, es imprescindible que los ingenieros apliquen un enfoque proactivo que integre las siguientes actividades:

1. **Elicitación** exhaustiva de las necesidades desde diversas dimensiones, incluyendo la identificación activa de los Requerimientos No Funcionales (RNF) de calidad y restricción.
2. **Análisis y Modelado** de estos requisitos para crear representaciones lógicas y coherentes, asegurando que se establece el alcance real del proyecto.<sup>[11]</sup>
3. **Validación y Verificación (V&V)** continua, ya que la **Validación** confirma que se está **construyendo el producto correcto** (satisfaciendo al cliente) y la **Verificación** asegura que se está **construyendo bien** (cumpliendo la especificación).<sup>[10]</sup>
4. **Gestión de Requerimientos**, la cual es esencial debido a la **volatilidad** y los cambios inevitables. Esta gestión, apoyada en la **Trazabilidad**, permite **adoptar y controlar el cambio** sin que este degrade la calidad del sistema.<sup>[1]</sup>

La aplicación de un tratamiento sistemático y estructurado de los requisitos es ineludible. Solo a través de una correcta obtención, especificación y gestión se puede garantizar que el *software* satisfaga las necesidades del consumidor, tenga un desempeño apropiado y confiable, y **agregue valor** para todos los que lo utilizan. **Sin buenos requerimientos, no puede haber buen software.**

## Bibliografía

1. Serna M., E. (2021). *Gestión de la ingeniería de requisitos integrando principios del pensamiento complejo*. Editorial Universidad del Valle. <https://dialnet.unirioja.es/descarga/libro/877217.pdf>
2. Rodríguez, G. J. (2023, 31 enero). *Técnicas efectivas para la toma de requerimientos*. Northware. <https://www.northware.mx/blog/tecnicas-efectivas-para-la-toma-de-requerimientos/>
3. Rojo, S. del V. & Oliveros, A. (2014). *Elicitación y Especificación de Requerimientos No Funcionales para aplicaciones Web*. (Referencia páginas 1-3). <https://43jaiio.sadio.org.ar/proceedings/ASSE/02-08-941-2582-2-DR.pdf>

4. Robertson, S. & Robertson, J. (2012). *Mastering the requirements process: Getting requirements right*. (3ª ed.). Addison-Wesley. Páginas (4-5, 13).  
[https://www.researchgate.net/profile/Ogbonnaya-Jr-Akpara/publication/330811319\\_Requirements\\_Specification/links/5c549ab6a6fdccd6b5da65bf/Requirements-Specification.pdf](https://www.researchgate.net/profile/Ogbonnaya-Jr-Akpara/publication/330811319_Requirements_Specification/links/5c549ab6a6fdccd6b5da65bf/Requirements-Specification.pdf)
5. Wiegers, K. & Beatty, J. (2013). *Software requirements*. (3ª ed.). Microsoft Press.  
[https://api.pageplace.de/preview/DT0400.9780735679641\\_A23707999/preview-9780735679641\\_A23707999.pdf](https://api.pageplace.de/preview/DT0400.9780735679641_A23707999/preview-9780735679641_A23707999.pdf)
6. Cueva, S. & Sucunuta, M. (2014). *Ingeniería de Requisitos*. Universidad Técnica Particular de Loja.  
<https://gc.scalahed.com/recursos/files/r161r/w24792w/IRADS/INGRequisitos.pdf>
7. Briano, C. (2023). *Compilación de apuntes sobre conceptos fundamentales de la Ingeniería de Software: con una visión orientada a proyectos vinculados con las Ciencias Económicas*. Ciudad Autónoma de Buenos Aires.  
[http://bibliotecadigital.econ.uba.ar/download/libros/Briano\\_compilacion\\_apuntes.pdf](http://bibliotecadigital.econ.uba.ar/download/libros/Briano_compilacion_apuntes.pdf)
8. Wong, S. (2017). *Análisis y requerimientos de software: manual autoformativo interactivo*. Huancayo: Universidad Continental.  
[https://repositorio.continental.edu.pe/bitstream/20.500.12394/4281/1/DO\\_FIN\\_103\\_MAI\\_UC0939\\_2018.pdf](https://repositorio.continental.edu.pe/bitstream/20.500.12394/4281/1/DO_FIN_103_MAI_UC0939_2018.pdf)
9. Pradel, J. & Raya, J. (2013). *Introducción a la ingeniería del software*. Universitat Oberta de Catalunya.  
<https://openaccess.uoc.edu/server/api/core/bitstreams/fab2269c-d5ae-4e93-934a-b0d91601d0f7/content>
10. Sommerville, I. (2011). *Ingeniería de Software*. PEARSON EDUCACIÓN.  
[https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9\\_compressed.pdf](https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf)
11. Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. McGraw Hill.  
[https://www.javier8a.com/umb/analisis/Libro\\_Pressman\\_7.pdf](https://www.javier8a.com/umb/analisis/Libro_Pressman_7.pdf)
12. Cristiá, M. (2024). *Introducción a la Ingeniería de Requerimientos*. Facultad de Ciencias Exactas, Ingeniería y Agrimensura. <https://www.fceia.unr.edu.ar/~mcristia/publicaciones/ingreq-a.pdf>