

## INGRS1401 – Ingeniería en Requerimiento de Software (1)

Felipe Alfonso González L.

Continuidad en Ingeniería en Informática, IACC Chile., 2020-2021

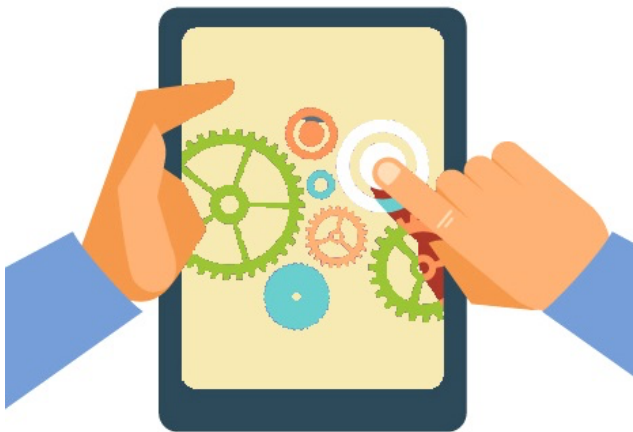
Instituto Superior de Artes y  
Ciencias de la Comunicación,  
IACC  
Av. Salvador 1318, Metro Santa  
Isabel, Providencia, Santiago.  
Chile.

*f.alfonso@res-ear.ch – felipe.alfonso.glz@gmail.com – <https://twitter.com/felipealfonsog>  
<https://glzengrg.com> - <https://freeshell.de/felipe> - <https://linkedin.com/in/felipealfonsog>*

*El ingeniero de requisitos recopila, analiza y verifica los requisitos del cliente para un sistema de software. Su propósito es proporcionar especificaciones correctas y requisitos de software.*

*La ingeniería de requisitos tiene como objetivo mejorar la comprensión y la definición de sistemas de software complejos.*

*Algunos autores la definen como la disciplina para desarrollar una especificación completa, consistente e inequívoca que servirá como un acuerdo común entre los usuarios del sistema, los ingenieros de software involucrados en el desarrollo y por lo tanto las funciones que el sistema funcionará.*



### *Objetivos de la ingeniería de software.*

En la actualidad, son muchos los procesos de desarrollo de software que existen. Con el pasar de los años, la Ingeniería de Software ha introducido y popularizado una serie de estándares para medir y certificar la calidad, tanto del sistema a desarrollar, como del proceso de desarrollo en sí. Se han publicado muchos libros y artículos relacionados con este tema, con el modelado de procesos del negocio y la reingeniería. Un número creciente de herramientas automatizadas han surgido para ayudar a definir y aplicar un proceso de desarrollo de software efectivo. Hoy en día la economía global depende más de sistemas automatizados que en épocas pasadas; esto ha llevado a los equipos de

desarrollo a enfrentarse con una nueva década de procesos y estándares de calidad.

Sin embargo, ¿cómo explicamos la alta incidencia de fallos en los proyectos de software? ¿Por qué existen tantos proyectos de software víctimas de retrasos, presupuestos sobregirados y con problemas de calidad? ¿Cómo podemos tener una producción o una economía de calidad, cuando nuestras actividades diarias dependen de la calidad del sistema?

Tal vez suene ilógico pero, a pesar de los avances que ha dado la tecnología, aún existen procesos de producción informales, parciales y en algunos casos no confiables.

La Ingeniería de Requerimientos cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

La razón principal para escoger este tema se fundamentó en la gran cantidad de proyectos de software que no llegan a cumplir sus objetivos. En nuestro país somos partícipes de este problema a diario, en donde se ha vuelto común la compra de sistemas extranjeros, para luego "personalizarlos" supuestamente a la medida de las empresas.

Tal "personalización", la mayoría de las veces, termina retrasando el proyecto en meses, o incluso en años. La problemática del año 2000 trajo como consecuencia una serie de cambios apresurados en los sistemas existentes; cambios que, desde mi punto de vista, no fueron bien planificados.

El reemplazo de plataformas y tecnologías obsoletas, la compra de sistemas completamente nuevos, las modificaciones de todos o de casi todos los programas que forman un sistema, entre otras razones, llevan a desarrollar proyectos en calendarios sumamente ajustados y en algunos casos irreales; esto ocasiona que se omitan muchos pasos

importantes en el ciclo de vida de desarrollo, entre estos, la definición de los requerimientos.

Estudios realizados muestran que más del 53% de los proyectos de software fracasan por no realizar un estudio previo de requisitos. Otros factores como falta de participación del usuario, requerimientos incompletos y el cambio a los requerimientos, también ocupan sitios altos en los motivos de fracasos.

Con este trabajo se pretende alcanzar los siguientes objetivos:

Resaltar la importancia que tiene la Ingeniería de Requerimientos dentro del ciclo de desarrollo.

Dar a conocer las diferentes alternativas que existen para identificar requerimientos.

Ayudar a comprender la diferencia que existe entre las diferentes técnicas utilizadas en la IR.

Minimizar las dudas que se tiene sobre los casos de uso.

Mostrar la utilización de herramientas CASE dentro de la administración de requisitos.<sup>i</sup>

En los sistemas que desarrollo en la asignatura anterior, debo hacer un vistazo hacia atrás, y comenzar a analizar con profundidad cuales son los elementos faltantes o que puedan haber sido elementos faltantes.

Al ingresar a cualquiera de mis sistemas no existían la posibilidad de que existieran CAPTCHA's para evitar que cualquier 'robot' o 'bot' intentara acceder al sistema o registrarse sin que el mismo sistema pudiera verificar que es un 'humano' en vez de un robot.

### **Problemática en cuestión**

Lo primero es identificar los objetivos que se persiguen en materia de calidad, a los que llamamos criterios de calidad, luego fijamos una forma para conocer si alcanzamos dichos objetivos, normalmente mediante índices numéricos a los que denominamos indicadores y, finalmente, establecemos un rango dentro del cual el nivel de calidad es el aceptable y que denominamos estándar de calidad. Es decir, cuál es el objetivo que pretendemos teniendo en cuenta aquellas características que mejor representan lo que deseamos lograr. Los Indicadores de Calidad son medidas cuantitativas que pueden usarse como guía para evaluar, controlar y valorar la calidad de las diferentes actividades.

Lo importante es tener un documento compartido, que signifique un elemento de discusión en reuniones TI.<sup>ii</sup>

### **Requerimientos sobre el sistema.**

En la especificación de requerimientos se deben tener en cuenta los requisitos del usuario o necesidades del cliente, esta es la base fundamental de un desarrollo de software, así

como también los requisitos que el software estar capaz de cubrir y cumplir con las necesidades planteadas. En conclusión, la ERS no es más que un informe técnico de cómo se elabora el software.

En mi software para ello el requerimiento, sería, elaborar un sistema CAPTCHA o utilizar la API o utilidad externa de software para crear lo que necesita el sistema en cuestión, ya mencionado.

### **Requerimientos funcionales**

Al ingresar debe tener una alternativa CAPTCHA. Validar mediante algún método externo o desarrollado internamente. Análisis constante del sistema, mediante analíticas.

#### *Técnicas de levantamiento de requerimientos software.<sup>iii</sup>*

#### *Técnicas para obtener requerimientos de software.*

1. - Análisis de documentación.
- 2.- Observación.
- 3.- Entrevistas.
- 4.- Encuestas o cuestionarios.
- 5.- Mesas de trabajo (Workshops)
- 6.- Tormenta de ideas.
- 7.- Historia del usuario.

*En la actualidad soy Head of Engineering en una empresa FANG, lo que nos hace buscar problemas en realidad y enfocarnos en ellos, en los problemas en sí. Generalmente esto no es usual en las empresas, ya que solo generan un documento muy simple sobre lo que debe funcionar, pero no de los problemas. Es una mirada totalmente diferente la que he logrado obtener en mi posición. Y me hace pensar en que antes de tener requerimientos en un documento, es bueno buscar un problema o los problemas a solucionar y rebuscar dentro de esos mismos problemas, más problemas.*

*De acuerdo a su desarrollo de sistema previo, describa los niveles de detalle de la especificación de requerimientos. Reconoce el nivel de detalle en la especificación de requerimientos.*

Lo primero es identificar los objetivos que se persiguen en materia de calidad, a los que llamamos criterios de calidad, luego fijamos una forma para conocer si alcanzamos dichos objetivos, normalmente mediante índices numéricos a los que denominamos indicadores y, finalmente, establecemos un rango dentro del cual el nivel de calidad es el aceptable y que denominamos estándar de calidad. Siempre habrá que

saber que, cómo, quién, cuándo, por qué y para qué se mide en el proceso de evaluación, que es condición indispensable para trabajar en Calidad.

Los Indicadores de Calidad son medidas cuantitativas que pueden usarse como guía para evaluar, controlar y valorar la calidad de las diferentes actividades. Es decir, la forma en la que se evalúa o se mide cada uno de los criterios.

Se construyen en base a la experiencia que se tenga de la actividad que se desarrolla, debiendo ser fáciles de establecer, enunciarse con objetividad, de la forma más sencilla posible, no implicar un elevado grado de dificultad en su interpretación, tratando de que todos los interesados entiendan y midan lo mismo y de idéntica forma, abarcar un amplio número de posibilidades, y deben resultar relevantes para la toma de decisiones. El Cuadro de Mando Integral es un conjunto de indicadores relacionados para los que se ha establecido un mecanismo de diseño y revisión sistemático y estructurado, buscando una gestión continua de los factores y procesos con influencia importante en las organizaciones y traduciendo los objetivos indicados en el Mapa Estratégico, que define la estrategia para la creación de valor y la calidad, en indicadores y metas a alcanzar. Los Estándares de Calidad se definen como el grado de cumplimiento exigible a un criterio de calidad.

En cuanto al desarrollo del sistema en asignaturas previas, se estableció lo siguiente “Se requiere desarrollar un sistema computacional, el cual consistirá en la digitalización de los viajes intercomunales que se realizan por parte de los vehículos institucionales, esto a través de la secretaría del Administrador Municipal, en donde se deben registrar las ciudades de destino, kilómetros estimados de recorrido, tiempos estimados de traslado, conductor designado, departamento que solicitante y horarios aproximados de salida y regreso al edificio consistorial. Permitiendo con esto que el Departamento de Administración Municipal pueda estimar los gastos de combustible y los kilómetros recorridos por los vehículos institucionales, y permitiendo asignarles sus respectivas mantenciones, a su vez la Dirección de Administración y Finanzas podrá confirmar los viáticos de los conductores y horas extras que se produzcan en dichos traslados de forma más dinámica.”

En esta etapa se deben describir los especificaciones de los requerimientos lo más detallada posible, contando los requisitos que necesita el cliente y sus necesidades, requisitos de software (para la aplicación planteada), en pocas palabras es un informe técnico.

*Documento de especificación del sistema (Nivel de detalles).<sup>iv</sup>*

- 1.1. Definición del problema.
- 2.2. Descripción funcional.
- 3.3. Restricciones Restricciones.
- 4.4. Diagramas de flujo de datos.
- 5.5. Modelo de Modelo de datos.
- 6.6. Diccionario de datos.
- 7.7. Casos de uso.
- 8.8. Documentos adicionales.

**Requerimientos:** Propiedades o restricciones Propiedades o restricciones determinadas de forma precisa determinadas

de forma precisa que deben satisfacerse. que deben satisfacerse. (En mi sistema la problemática es que no cuenta con sistema de verificación CAPTCHA)

**Requerimientos funcionales:** Expresan la naturaleza del funcionamiento del sistema (cómo interacciona el sistema con su entorno y cuáles (cómo interacciona el sistema con su entorno y cuáles van a ser su estado y funcionamiento). – En este caso la naturaleza es la verificación de que no sea un Bot.

**Requerimientos no funcionales:** Restricciones sobre el espacio de posibles soluciones.

**Rendimiento del sistema:** Fiabilidad, tiempo de respuesta, disponibilidad...

**Interfaces:** Dispositivos de E/S, usabilidad, interoperabilidad...

**Proceso de desarrollo:** Estándares, herramientas, plazo de entrega...

*Requerimientos funcionales y no funcionales.*

**Los requisitos funcionales definen qué debe hacer un sistema.**

**Los requisitos no funcionales definen cómo debe ser el sistema.**

*En concreto,*

Todo lo anterior significa, tomar en cuenta las siguientes:

1. Definición del problema.
- 2.2. Descripción funcional (lista de requerimientos funcionales).
- 3.3. Restricciones (requerimientos no funcionales).
- 4.4. Diagramas de flujo de datos.
- 5.5. Modelo de datos (diagrama E/R, CASE\*(diagrama E/R, CASE\*Method o diagrama de clases UML).
- 6.6. Diccionario de datos.
- 7.7. Casos de uso..
- 8.8. Documentos adicionales (p.ej. modelos de informes y formularios).

*Es importante acotar que siempre existirán costos en hora hombre que se deben tomar en cuenta, o tiempos, etc.*

*Según su consideración, ¿Cuáles son los criterios de calidad aplicados en el desarrollo de sistema?, Justifique. Reconoce la importancia de la ingeniería de requerimientos en el diseño de software.*

*La Evaluación del Sistema*

Se lleva a cabo para identificar puntos débiles y fuertes del Sistema implantado. La evaluación ocurre a lo largo de cualquiera de las siguientes cuatro dimensiones:

*Evaluación operacional*

Es el Momento en que se evalúa la manera en que funciona el Sistema, esto incluye su facilidad de uso, Tiempo de respuesta ante una necesidad o proceso, como se adecuan los formatos en que se presenta la Información, contabilidad global y su nivel de Utilidad.

## *Impacto Organizacional*

Identifica y mide los beneficios operacionales para la Empresa en áreas tales como, Finanzas (Costos, Ingresos y Ganancias), eficiencia en el desempeño laboral e impacto competitivo, Impacto, rapidez y organización en el flujo de Información interna y externa.

## *Desempeño del Desarrollo*

Es la evaluación del Proceso de desarrollo adecuado tomando en cuentas ciertos criterios como, Tiempo y esfuerzo en el desarrollo concuerden con presupuesto y estándares y otros criterios de Administración de Proyectos. Además se incluyen la valoración de los métodos y herramientas utilizados durante el desarrollo del Sistema.

## *Prueba de Sistemas*

Dependiendo del tamaño de la Empresa que usara el Sistema y el riesgo asociado a su uso, puede hacerse la elección de comenzar la operación del Sistema solo en un área de la Empresa (como una Prueba piloto), que puede llevarse a cabo en un Departamento o con una o dos personas. Cuando se implanta un nuevo sistema lo aconsejable es que el viejo y el nuevo funcionen de manera simultanea o paralela con la finalidad de comparar los resultados que ambos ofrecen en su operación, además dar tiempo al personal para su entrenamiento y adaptación al nuevo Sistema.

Durante el Proceso de Implantación y Prueba se deben implementar todas las estrategias posibles para garantizar que en el uso inicial del Sistema este se encuentre libre de problemas lo cual se puede descubrir durante este proceso y llevar a cabo las correcciones de lugar para su buen funcionamiento.

Desdichadamente la evaluación de Sistemas no siempre recibe la atención que merece, sin embargo cuando se lleva a cabo de manera adecuada proporciona muchas informaciones que pueden ayudar a mejorar la efectividad de los esfuerzos de desarrollo de aplicaciones futuras.<sup>v</sup>

## *Evaluación operacional.*

Es importante considerar siempre, la complejidad de uso del problema a solucionar (niveles), tiempos de respuestas, analíticas, y claridad de lo que se necesita realizar o se realizara.

## *Impacto organizacional.*

Dentro de los que debernos tomar en cuenta en un grupo de desarrollo en conjunto con la gerencia es si se ajusta al presupuesto, si ayuda al sistema en si, y si permite que el sistema sea mas robusto, optimo e inteligente.

Hoy en día las compañías de todo el mundo industrializado reconocen que la calidad del producto se traduce en ahorro de costos y en una mejora general. La industria de desarrollo de software no es la excepción, por lo que en los últimos años se han realizado intensos trabajos para aplicar los conceptos de calidad en el ámbito del software. Hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de calidad. El problema es que la mayoría de las características que definen al software no se pueden cuantificar fácilmente; generalmente, se establecen de forma cualitativa, lo que dificulta su medición, ya que se requiere establecer métricas que permitan evaluar cuantitativamente cada característica dependiendo del tipo de software que se pretende calificar.

En este sentido se han realizado muchos trabajos que establecen propuestas para el establecimiento de los factores cualitativos que afectan la calidad del software. Entre los principales están los factores de calidad de McCall [1][4] y aquellos propuestos por Hewlett-Packard (FURPS: Functionality Usability, Reliability; Performance, Supportability) [4]. Además se han hecho varios intentos por estandarizar los mecanismos de evaluación de calidad del software. Entre los principales están la familia de normas ISO 9000 (en especial la ISO 9001 y la ISO 9003-2) [5], el modelo de niveles madurez CMM (Capability Maturity Model) [7], el estándar para el aseguramiento de planes de calidad del IEEE 730:1984 [7], el plan general de garantía de calidad del Consejo Superior de Informática MAP [7] y la norma ISO/IEC 9126 [3], que es objeto de este estudio.

La ISO, bajo la norma ISO-9126, ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 con el nombre de "Information technology-Software product evaluation: Quality characteristics and guidelines for their use", en el cual se establecen las características de calidad para productos de software. El estándar ISO-9126 [7] establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software. La tabla 1 muestra la pregunta central que atiende cada una de estas características.

<i>Características</i>	<i>Pregunta central</i>
<i>Funcionalidad</i>	¿Las funciones y propiedades satisfacen las necesidades explícitas e implícitas; esto es, el qué...?
<i>Confiabilidad</i>	¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo?
<i>Usabilidad</i>	¿El software es fácil de usar y de aprender?
<i>Eficiencia</i>	¿Es rápido y minimalista en cuanto al uso de recursos?
<i>Mantenibilidad</i>	¿Es fácil de modificar y verificar?
<i>Portabilidad</i>	¿Es fácil de transferir de un ambiente a otro?

Tabla 1. Características de ISO-9126 y aspecto que atiende cada una.



detalla cada una de las características que establece el estándar ISO-9126.

**C1. Funcionalidad** En este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. Para este propósito se establecen los siguientes atributos:

- **Adecuación.** Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.
- **Exactitud.** Este atributo permite evaluar si el software presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
- **Interoperabilidad.** Permite evaluar la habilidad del software de interactuar con otros sistemas previamente especificados.
- **Conformidad.** Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
- **Seguridad.** Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.

#### C2. Confiabilidad

Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. Las subcaracterísticas que el estándar sugiere son:

- **Nivel de Madurez.** Permite medir la frecuencia de falla por errores en el software.
- **Tolerancia a fallas.** Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- **Recuperación.** Se refiere a la capacidad de restablecer el nivel de operación y recuperar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.

#### C3. Usabilidad

Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.

- **Comprensibilidad.** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructuralógica del sistema y los conceptos relativos a la aplicación del software.
- **Facilidad de Aprender.** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
- **Operabilidad.** Agrupa los conceptos que evalúan la operación y el control del sistema.

#### C4. Eficiencia

Esta característica permite evaluar la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados. Los aspectos a evaluar son:

- **Comportamiento con respecto al Tiempo.** Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos.
- **Comportamiento con respecto a Recursos.** Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

#### C5. Mantenibilidad

Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al software, ya sea

por la corrección de errores o por el incremento de funcionalidad. En este caso, se tienen los siguientes factores:

- **Capacidad de análisis.** Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
- **Capacidad de modificación.** Mide el esfuerzo necesario para modificar aspectos del software, remover fallas o adaptar el software para que funcione en un ambiente diferente.
- **Estabilidad.** Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al software.
- **Facilidad de Prueba.** Se refiere al esfuerzo necesario para validar el software una vez que fue modificado.

#### C6. Portabilidad

En este caso, se refiere a la habilidad del software de ser transferido de un ambiente a otro, y consideramos los siguientes aspectos:

- **Adaptabilidad.** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.
- **Facilidad de Instalación.** Es el esfuerzo necesario para instalar software en un ambiente determinado.
- **Conformidad.** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.
- **Capacidad de reemplazo.** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.

#### Análisis de lo anterior.

El mundo globalizado exige cada vez más la aplicación de estándares internacionales que garanticen la calidad de los productos. Por esta razón, es necesario que todo aquel que se dedica al desarrollo de software incluya en sus procesos, estándares de calidad que permitan certificarse en alguno de los modelos. Aquí se ha presentado un estándar, el ISO-9126, el cual establece una guía para la evaluación de la calidad del software, sin embargo es necesario que cada empresa dedicada a producir software trabaje en establecer su modelo de calidad que le permita valorar el nivel de excelencia de sus productos, en el que deberán incluirse instrumentos de medición que permitan calificar cuantitativamente cada una de las características aquí presentadas. Es importante mencionar, que dependiendo de los distintos tipos de aplicaciones las métricas podrán variar, ya que aunque las características expuestas son comunes a la totalidad de los productos, cada software particular requiere una evaluación específica.

#### Análisis y conclusión final.

Tener claro los requisitos de una propuesta le da un valor agregado al grupo de trabajo, porque la información verificada, permite documentar de manera efectiva el desarrollo del proyecto.

Al aplicar la ingeniería de software, las especificaciones de los requisitos son completa y correcta, para satisfacer las necesidades finales.

## *Tipos de documentación del proyecto. Ingeniería de Proyectos y requerimiento de software.<sup>vii</sup>*



Los tipos de documentación de un proyecto son muy importantes y pueden estudiarse de la siguiente manera:

- Cómo estructurar la documentación de los proyectos.
- Qué documentos básicos son vinculantes.
- El modo de justificar la adopción de decisiones dentro de la memoria.
- Dónde ubicar el material que, siendo de interés, obstaculizaría la normal lectura de la memoria (anexos).
- La necesidad de determinar las características que habrán de poseer los recursos y procesos necesarios para el desarrollo del proyecto (pliego de condiciones).
- La forma de presentar ordenadamente la representación gráfica del resultado buscado del proyecto (planos).
- Cómo evaluar y presentar, en el estado de mediciones, la cantidad de recursos necesarios para el desarrollo del proyecto.
- El modo en el que se desarrolla una estimación monetaria del coste de la obra (presupuesto).
- Dónde ubicar aquellos estudios de obligada presentación ante la administración (estudios con entidad propia).
- Qué es un informe técnico.
- Los distintos tipos de informes técnicos que existen y las diferencias que existen entre estos.
- Qué son y cuándo se utilizan las memorias valoradas.

### *Organización documental del proyecto*

En el sentido clásico, es decir, según el Decreto de 19 de octubre de 1961, sobre «Tarifas de honorarios de ingenieros en trabajos a particulares», los documentos que debe incluir un proyecto son cuatro:

- Memoria
- Planos
- Pliego de condiciones
- Presupuesto

Sin embargo, no todos ellos tienen la misma fuerza de obligar legalmente, ya que, aunque todos son informativos, los planos, el pliego de condiciones y el apartado «cuadro de precios» del presupuesto son vinculantes y, en consecuencia, en caso de discrepancia entre los distintos documentos, su información prevalece sobre el resto. Por otro lado, y de cara a la Administración Pública, ese carácter vinculante se manifiesta en que el compromiso de

la Propiedad ante ella es realizar lo que en esos documentos aparece como definitorio del proyecto. Debido a la reducida normativa existente durante más de 40 años, la información necesaria para definir la estructura y elaboración del documento de un proyecto ha sido algunas veces escasa, otras veces contradictoria, etc.; y casi siempre, se ha fundamentado en la bibliografía docente y en algunas pocas normas que definían la estructura o simbología de los documentos. En el año 2014, con la aprobación de la norma UNE 157001:2014, «Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico», se establecieron las consideraciones generales que permiten precisar las características que deben satisfacer los proyectos de productos, obras y edificios, instalaciones y servicios, para que sean conformes al fin a que están destinados. Con esta nueva norma UNE, de los cuatro documentos de un proyecto se pasa<sup>viii</sup> a siete y se representan en el siguiente orden:

- Índice
- Memoria
- Anexos
- Planos
- Pliego de condiciones

*Según la norma UNE 157001:2014, se seguirán. Las siguientes normas de presentación:*

- Se ordenarán los documentos del proyecto («Índice», «Memoria», etc.) según lo requerido por la tipología del proyecto, pudiendo agruparse en distintos volúmenes.
- En función de la normativa legal aplicable, a criterio del proyectista o dependiendo del tipo de proyecto, cabe la posibilidad de no incluir la totalidad de los mismos, o añadir otros.
- Cada uno de los documentos del proyecto se puede descomponer en otros documentos. Se estructurarán en capítulos y apartados y se numerarán de acuerdo con lo indicado en la norma.
- En todas las páginas del proyecto debe figurar el título, código o referencia de identificación, tipo de documento («Índice», «Memoria», etc.), número de página, número total de páginas y, si el proyecto dispone de varios volúmenes, en cada uno de ellos se debe incluir, además, su título y número de volumen.
- El proyecto debe iniciarse con la siguiente información:
  - El título del proyecto y su código o referencia de identificación.
  - El nombre del organismo u organismos a los que se dirige el proyecto.
  - El nombre, apellidos y documento de identidad del promotor del proyecto, ya sea en nombre propio o en representación de otra persona física o jurídica, cuyos datos también deben figurar junto con cualquier otro identificador que pudiera existir y cuya revelación sea legalmente procedente.
  - Si el proyecto contempla un emplazamiento geográfico concreto, se definirá dicho emplazamiento (dirección y población) y, si procede, sus coordenadas UTM (Universal Transverse Mercator).
  - El nombre de la organización, razón social o persona jurídica que ha recibido el encargo de elaborar el proyecto, así como su C.I.F., dirección, teléfono, fax, correo electrónico o cualquier

otro identificador profesional que pueda aparecer o existir, salvo aquellos casos cuya publicidad no sea legalmente procedente.

- El nombre y apellidos del proyectista, su titulación, y, en su caso, colegio o entidad a la que pertenece, documento de identidad, domicilio profesional, teléfono, fax, correo electrónico o cualquier otro identificador profesional que pueda aparecer o existir, salvo aquéllos casos cuya publicidad no sea legalmente procedente.

### Ingeniería de detalle.<sup>ix</sup>

importancia de los distintos componentes de la ingeniería básica hace imprescindible un estudio profundo de todos sus aspectos. Es necesaria su discusión y su establecimiento como básicos y «casi» inamovibles en el futuro. Dada la importancia de este aspecto para la Empresa de Ingeniería y la Propiedad, es necesaria una aprobación por parte de todos los interesados de la mayoría de los documentos que integran la ingeniería básica. La finalización de la ingeniería básica en sus distintos aspectos supone un punto de inflexión en la realización del proyecto semejante al que se discutió al finalizar los estudios previos. Es la última ocasión de modificar, de forma económica y racional, cualquier característica del proyecto; y de llegar, incluso, a su suspensión si es necesaria. Por todo ello, una vez estudiada, discutida y aprobada, debe reflejarse en un soporte documental adecuado que permita iniciar las etapas siguientes con las responsabilidades claramente asumidas de cada una de las partes.

### Definición de la ingeniería de detalle.

Al llegar a este punto y siguiendo el desarrollo normal del proyecto, se tiene ya una organización en marcha y una ingeniería básica, debidamente aprobada y disponible para continuar el trabajo. Es en este momento cuando se pasa a la ingeniería de desarrollo. Ésta tiene como etapas importantes las siguientes:

- Ingeniería de detalle.
- La gestión de compras.
- Construcción y montaje.
- Puesta en servicio.

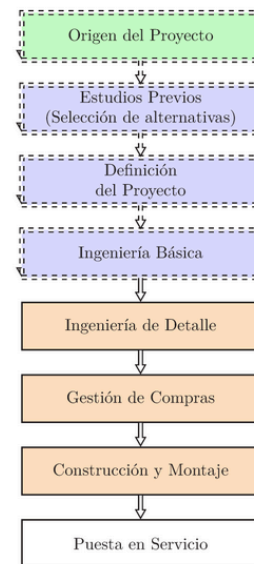


Figura 12.1: Esquema general de fases del proyecto

Se suele conocer mas la ingeniería de detalle en Ingeniería de Obras de Construcción, pero hay comentarios papers que indican que también se puede aplicar a las TIC.

### Ingeniería del software<sup>x</sup>

¿Qué es la ingeniería del software?

Un sistema de software, denominado también aplicación o simplemente software, es un conjunto integrado de programas que en su forma definitiva se pueden ejecutar, pero comprende también las definiciones de estructuras de datos (por ejemplo, definiciones de bases de datos) que utilizan estos programas y también la documentación referente a todo ello (tanto la documentación de ayuda en el uso del software para sus usuarios como la documentación generada durante su construcción, parte de la cual también servirá para su mantenimiento posterior).

### El software como producto industrial

Un software no es una obra de arte, sino un producto de consumo utilitario y masivo; para una empresa o trabajador autónomo, el software es un medio auxiliar que interviene de manera mas o menos indirecta, pero a menudo imprescindible, en su gestión y cada vez mas en su proceso productivo; también existe, como todos sabemos, un consumo privado de software. Por tanto, se puede considerar plenamente como un producto industrial. Por ejemplo... los bancos, las industrias de fabricación en serie, las empresas de comercio electrónico, etc. actualmente no podrían funcionar sin software. Sin embargo, es un producto industrial con algunas características especiales. En primer lugar, es mucho mas un producto singular que un producto que se fabrique en serie (aunque algunos softwares tienen muchos miles de usuarios e, incluso, millones), ya que, si bien existe —y no siempre— producción en serie de copias del software, esta es una actividad muy poco importante dentro del conjunto de su proceso productivo y relativamente sencilla.

## ¡La producción de software se parece a la construcción!

Desde cierto punto de Vista, la producción de software se parece a la construcción de Viviendas o edificios industriales, por ejemplo, en el hecho de que cada producto es diferente y su elaboración se basa en un proyecto específico (en el caso de producción en serie, lo que se proyecta es un prototipo del producto, y no cada unidad que se produce).

Otras características del software son, como señala Pressman, que no se estropea por el uso ni por el paso del tiempo. Si finalmente se tiene que sustituir es porque se ha quedado tecnológicamente anticuado o inadaptado a nuevas necesidades o porque ha llegado a resultar demasiado caro mantenerlo.

```
function GifAnim(layer, imgName, imgSeries, end, speed, startFrame)
{
  this.layer = layer
  this.imgName = imgName
  this.frame = new Array()
  for (var i=0; i<=end; i++) this.frame[i] = imgSeries+i
  this.end = end
  this.speed = speed
  this.active = false
  this.count = (StartFrame) ? startFrame : 0
  this.obj = imgName + "GifAnim"
  eval(this.obj + "=this")
  this.play = GifAnimPlay
  this.run = GifAnimRun
  this.stop = GifAnimStop
  this.goToFrame = GifAnimGoToFrame
}
```

## Los diagramas de implementación.

Los diagramas de implementación, a diferencia de los estáticos y de los dinámicos, no describen la funcionalidad del software, sino su estructura general con Vistas a su construcción, ejecución e instalación. Son dos:

- El diagrama de componentes, que muestra cuales son las diferentes partes del software.
- El diagrama de despliegue, que describe la distribución física de las diferentes partes del software en tiempo de ejecución. Se utilizan en el diseño y la implementación.

## El diagrama de componentes

El diagrama de componentes describe la descomposición física del sistema de software (y, eventualmente, de su entorno organizativo) en componentes, a efectos de construcción y funcionamiento. La descomposición del diagrama de componentes se realiza en términos de componentes y de relaciones entre los mismos. Los componentes los componentes identifican objetos físicos que hay en tiempo de ejecución, de compilación o de desarrollo, y tienen identidad propia y una interfaz bien definida.

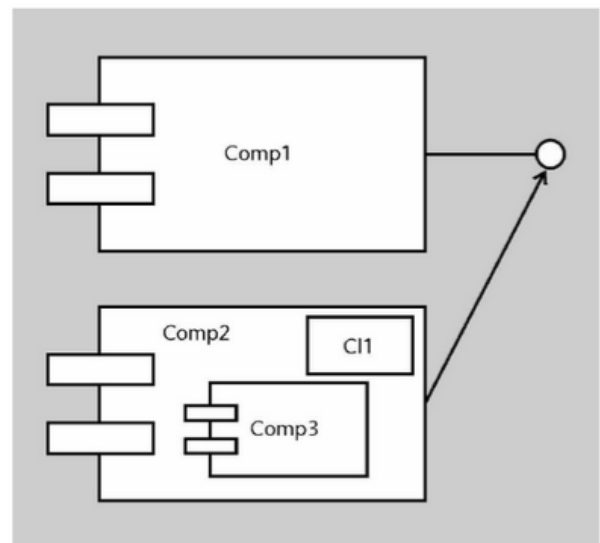
Los componentes incluyen código en cualquiera de sus formatos (código fuente o ejecutable), DLL, imágenes, pero también pueden ser documentos manuales cuando se describen partes no informatizadas de un sistema de información.

## Relaciones entre componentes

En un diagrama de componentes se muestran las diferentes relaciones que se pueden establecer entre componentes y otros componentes, objetos o procesos (objetos activos). En el caso de componentes no informáticos, el significado de la relación es que un componente utiliza la información contenida en el otro. En el caso de

componentes de software, se distinguen dos tipos de relaciones, las relaciones en tiempo de desarrollo y las relaciones de llamada: - Las relaciones en tiempo de desarrollo son asociaciones entre componentes que modelan dependencias las cuales se tendrán en cuenta en tiempo de compilación o en tiempo de enlace. - Las relaciones de llamada son asociaciones entre componentes que sirven para modelar llamadas entre componentes, es decir, que un componente —el cliente— utilice servicios de otro —el proveedor—. En tiempo de desarrollo, las relaciones de llamada se establecen entre componentes de tipo, y se representan en los diagramas de componentes. En tiempo de ejecución, se producen entre dos componentes de instancia y se representan en los diagramas de despliegue. Además, un componente puede tener relaciones de agregación y composición con otros componentes y con objetos. En el caso de la composición, los componentes de un componente se pueden representar dentro del mismo.

Ejemplo de diagrama de componentes Observad el siguiente diagrama de componentes:



El componente CDmpZ utiliza el Cumpl y contiene el componente Comp3 y un objeto de la clase C11.

## Pasos de la recogida y documentación de requisitos

Los pasos de la recogida y documentación de requisitos son los siguientes:

- 1) Conocimiento del contexto del futuro software.
- 2) Recogida y clasificación de los guiones.
- 3) Identificación de los actores.
- 4) Identificación de los casos de uso a partir de los guiones.



S) Identificación de relaciones entre casos de uso (extensión, inclusión, especialización).

6) Identificación de las relaciones de especialización entre actores.

7) Documentación de los casos de uso.

*El contexto del software* Los desarrolladores de software que se encargarán de recoger los requisitos generalmente tendrán formación y experiencia informáticas, pero no conocerán la actividad profesional de los usuarios. Si es así, será conveniente que adquieran cierto conocimiento desde el punto de Vista organizativo y, como consecuencia, de la terminología que se utiliza, - esto es lo que denominamos contexto del software. Hay dos maneras de describir el contexto de un software:

- el modelo del dominio, que es la manera simplificada, y
- el modelo del negocio, que es una modalidad más detallada.

En cualquier caso, hay que elaborar un glosario de los términos más utilizados. Incluso si no se realiza ninguno de los dos modelos, conviene confeccionar el glosario.

## Introducción

El propósito de la obtención de requisitos es adquirir un problema relevante para el que se utilizará para producir un software formal necesario para resolverlo.

Debido a que los problemas son desacuerdos entre dos, esta premisa muestra una vez más la importancia de la comunicación entre desarrolladores y clientes; esta comunicación con el cliente depende de la comprensión de sus necesidades. En el final de la fase de análisis de necesidades, puede tener un conocimiento profundo del área del problema.

---

La elicitación de requerimientos es un proceso de manifestación de los requerimientos para una aplicación o software que a través de la comunicación con los clientes o usuarios del sistema que tengan algún tipo de conocimiento sobre el proyecto a desarrollar

Uno de los principales efectos del proceso de elicitación es la sucesión de requerimientos que el equipo de desarrollo de software deberá utilizar, para la creación un producto eficiente y eficaz. Además de este resultado esperado el proceso debe ofrecer una sucesión de salidas intangibles.

Un buen proceso de elicitación facilita a los usuarios para entender de mejor forma que es lo que necesitan, si existe alguna restricción y alternativas, sus ventajas y desventajas. Para que exista un buen proceso de elicitación, es importante la visión o punto de vista del Ingeniero de Requerimientos, este ayudara a los usuarios a simplificar el

problema o petición, es decir, lo que realmente desea y necesita el usuario, resolver un problema que sea factible, entender y obtener la confianza del usuario en conjunto con su cooperación y a ganar conocimiento sobre el dominio del problema o solicitud.

Para una buena elicitación se tienen que considerar los siguientes puntos:

### Perspectiva de Negocio:

Los stakeholders relacionados con el negocio y con marketing deben identificar las metas y los objetivos en alto nivel de la organización u empresa, también como responder a preguntas: ¿de cómo? ¿por qué se está tratando de buscar una solución a la problemática?, el enfoque de clientes al que va destinado el producto, qué necesidades del negocio se tendrán que compensar, y cuáles serán las métricas para identificar si el proyecto ha sido exitoso.

### Perspectiva del Usuario:

Los requerimientos más característicos son los del usuario. El es el encargado debe responder las preguntas sobre qué problemas requieren ser solucionados de acuerdo con su manejo dentro del negocio y cómo se interacciona con el producto.

### Perspectiva Técnica:

Los desarrolladores entregan la perspectiva técnica necesaria para responder preguntas sobre cómo alcanzar los objetivos del proyecto, mejorar que puedan aportar a desarrollo óptimo

## Técnicas de Elicitación Propuestas.<sup>xi</sup>

Análisis de documentación existente: En esta ocasión no se revisaron documentación existente, pero si se hizo análisis de una aplicación que está operativa online, que cumple similares funciones.

Glosario en detalle<sup>xiixiii</sup>.

**Término/Sigla**  
**Descripción**

**Crédito Hipotecario** Es un préstamo a mediano o largo plazo que se otorga para la compra, ampliación, reparación o construcción de vivienda, la compra de terrenos, oficinas o locales o para provisión gratuita. ampliación, reparación o construcción de vivienda, compra de terrenos, oficinas o locales o por provisión gratuita.

**Mutuo Endosable** Se financia mediante un instrumento emitido por la denominada "factura hipotecaria".

Estos pueden ser negociados por el banco en la bolsa de valores o adquiridos por el propio banco o un tercero obteniendo así los recursos que financian el crédito concedido al deudor.

El precio que se obtiene de la venta de estos boletos en función de las condiciones del mercado, de manera que se puede generar un entre el valor del boleto y el precio al que se negocia.

En el acto de hipoteca o crédito mutuo, la parte contratante será responsable de esta diferencia debe especificarse.

En caso de que resulte que es el deudor quien debe pagarlo, por lo tanto, se recomienda que antes de realizar la transacción, busque en el mercado el valor al que se intercambiaron facturas similares. en los días anteriores a su propia transacción.

**Mutuo no Endosable** Se trata de un préstamo en pesos o en unidades de desarrollo y se sustenta en un acto contractual, el cual negociado en el mercado mediante un anexo, permite al titular del contrato de pago a terceros para su financiación.

El solicitante de crédito recibe el monto aprobado y no se genera diferencia así pueden ocurrir los préstamos otorgados con letras hipotecarias.

Este tipo de crédito puede ser transferido por medio de un corredor, el cual está registrado en la escritura respectiva.

No obstante, lo anterior, la administración del crédito queda en el banco, por lo que el canal de comunicación del deudor universal sigue siendo éste.

**Letras de Crédito** En este caso, el Banco financia el préstamo con recursos propios, pero a diferencia del anterior, este no se puede transferir por modificación.

Su regulación se limita a lo pactado en el contrato respectivo y a lo establecido en la Ley 18.010 de operaciones de crédito monetario.

**Tasa Fija** Los préstamos con tasa de interés fija son aquellos en que la tasa de interés aplicada al crédito permanece invariable a lo largo del plazo pactado.

**Tasa Variable** Los préstamos a tasa variable son aquellos para los que la tasa de interés cambia de acuerdo con un punto de referencia.

En consecuencia, el dividendo a pagar varía a lo largo de la evolución del préstamo ya que el interés del préstamo no es el mismo durante todo el período.

**Tasa Mixta** Los préstamos de tasa mixta son una combinación de la primera, es decir, una hipoteca que tiene una tasa fija para una primera período, luego una tasa de interés.

### 3. *Observación*

Perspectiva del producto<sup>xiv</sup>:

*Permitir a los interesados ingresar a un sitio web para poder realizar la simulación de un Crédito Hipotecario, permitiendo la comparación con múltiples entidades bancarias e instituciones financieras, además de distintos tipos y condiciones crediticias.*

Funciones del producto:

Carga de parámetros a página principal, en este punto se realizará la carga inicial de la página con los datos requeridos. Desplegar entidades bancarias e instituciones financieras, se desplegará una lista de instituciones permitiendo su múltiple selección.

Ingresos de datos del usuario para simulación, el usuario deberá proporcionar datos como: valor propiedad, tipo de crédito, cantidad de años y valor a solicitar.

Resultado visual de simulación, este desplegará el resultado y el cálculo de los datos entregados por el cliente.

### Análisis Social:

Es una aproximación simple a la elicitación de requerimientos que transporta la importancia desde la formalización de los requerimientos escritos hacia la conversación. Las historias de usuario deberán ser escritas por el usuario, dando énfasis en aquellas funcionalidades que el sistema o aplicación deberá realizar. Generalmente, estas historias suelen ser un conjunto reducido de frases u oraciones largas.

### Workshops:

Nos entrega la oportunidad para compartir, clarificar y ajustar las perspectivas individuales sobre los requerimientos del negocio.

Si bien existen diferentes elementos a elicitar durante la fase de requerimientos, metas del sistema, stakeholders, objetivos, tareas,

factibilidad, riesgos. Según la literatura sugiere que la elicitación de requerimientos se realice de distintas maneras, pudiendo seguir una o más metodologías disponibles; sin embargo, la habilidad de implementación de este proceso puede presentar más variables debido a la dificultad de formalizar indicadores directos o indirectos (métricas) para la evaluación y monitoreo del proceso.

### **Prototipo**

El prototipo es la representación o visualización de las partes real del sistema. El prototipo está diseñado en las primeras etapas de la ejecución del proyecto. Proporciona la idea general de las funciones del sistema actual y el flujo de trabajo. Los prototipos se utilizan para recopilar los requisitos de los usuarios mediante la presentación de las funciones en una interfaz gráfica de usuario basado en el sistema. Un prototipo representa el producto real, tanto en sentido funcional como gráfico. Proporciona la flexibilidad para los usuarios y las partes interesadas a trabajar con la versión inicial del producto para entender el sistema y pensar en las necesidades adicionales que el usuario haya olvidado mencionar. Los prototipos es uno de los métodos más caros.

### **Reúso de Requerimientos**

En el ámbito de la ingeniería de software la reutilización de los requerimientos del sistema actual es el método común de obtención de requerimientos. Con los actuales conocimientos para desarrollar el nuevo producto, se tienen muchas ventajas que incluyen bajo costo y menos tiempo. Aunque cada producto tiene su propio tipo de partes interesadas y usuarios, todavía hay varias situaciones que la reutilización de los requerimientos lleva a cabo.

### **Escenarios**

Los escenarios son ejemplos de las sesiones de interacción y estos consisten en descripciones de las acciones secuenciales. Los escenarios son útiles porque a los usuarios finales y otras partes interesadas del sistema les resulta más fácil relacionarse con ejemplos

de la vida real, en lugar de descripciones abstractas de las funciones. Los escenarios deben incluir al menos los siguientes tipos de descripciones.

### **Glosario Simplificado.** <sup>xv</sup>

#### Atributo de Requerimiento .

Cada uno de los atributos que se especifican para un requerimiento de software.

#### **Diagrama de Actividad.**

Diagrama que hace parte del lenguaje UML. Permite modelar el flujo entre un conjunto de objetos que cooperan entre sí. Son similares a los diagramas de flujo de otras metodologías diferentes a la orientada a objetos.

#### **Requerimientos de Software.**

Documento, o conjunto de documentos en el cual se consignan de manera preliminar a la fase de especificación los requerimientos de software de un sistema.

#### **Casos De Uso.**

Documento que contiene la especificación de casos de uso definidos para un sistema. Así mismo, puede contener un diagrama de casos de uso.

#### **Modelo Conceptual.**

Modelo que define vistas que representan la organización de los componentes, agentes o elementos de software que participan para lograr la funcionalidad requerida por el sistema.

#### **Stakeholder.**

Persona interesada o involucrada en el desarrollo de un sistema, bajo una perspectiva. Esta puede ser económica o relacionada otro beneficio por el desarrollo del sistema.

#### **UML (Unified Modeling Language).**

Lenguaje de Modelamiento Unificado. Es un lenguaje para especificar, construir, visualizar y documentar los artefactos o ítems de un sistema o software orientado a objetos

#### Análisis.

La obtención de los requisitos está

dedicada a la descripción de los cuales conducirán a los ingenieros de software a desarrollar soluciones de software de calidad.

El modelo tradicional puede ignorar aspectos importantes debido a que está especificado en los problemas ambientales de lo que hace, se requiere comunicación entre los clientes y todas las personas involucradas en el software.

En la medida que independicemos la fase de diseño de la de análisis, daremos más libertad a la implementación y obtendremos un mejor producto. Que no entren en conflicto con otros requisitos. Describe de manera completa, sin deficiencias ni necesidad adicional de información, el requisito. También es necesario especificar el nivel y alcance del requisito, no es lo mismo que el sistema por diseño sea capaz de cumplir el requisito, que verificar de manera continuada y rigurosa que el requisito se cumple durante la explotación del sistema.

La definición incluye un único requisito. Lo que luego nos puede ayudar a auditar y validar cada requisito de manera única y sencilla. Esto incluye que sean posibles bajo los análisis de riesgo o coste-beneficio.

### **Los requisitos pueden ser funcionales o no funcionales**

Funcionales. No funcionales. Se podría decir que los requisitos funcionales representan la parte del mundo real que toca al sistema y los no funcionales la parte tecnológica. El análisis completo de los requisitos funcionales es relativamente sencillo.

Los cambios, añadidos y eliminaciones de requisitos funcionales suelen tener impacto en el alcance del proyecto. Sin embargo, hacer un análisis completo de los requisitos no funcionales es más complicado. Si cambiamos el diseño, muy posiblemente el requisito cambie o pierda consistencia. Si un requisito no funcional puede cambiarse por otro o cambia si cambiamos el diseño, es muy probable que estemos hablando de una consecuencia del diseño de la solución, y no de un requisito como tal.

Este aspecto es importante, porque los requisitos no deben condicionar la implementación. Desgraciadamente un requisito no funcional pasado por alto puede tener efectos catastróficos en un diseño o implementación ya iniciados. Esta lista, como checklist que es, tiene como único objetivo facilitar que el análisis de requisitos sea completo y evitar que se nos pase por alto algún aspecto que podría ser útil a nuestro análisis. La gran mayoría de los requisitos de esta lista no van a aplicar ni tener sentido para el problema que estamos analizando, pero lo importante es que no se nos pase aquel requisito que sí es relevante, beneficioso o imprescindible.

Por último, el conjunto de los requisitos no funcionales tienen unos efectos en el reparto de la carga de trabajo de implementación que, por lo general, son mayores que el de los requisitos funcionales. Cada diferente requisito no

funcional debe abordarse como una solución separada, heterogénea y poco acoplada con las de otros requisitos, con las consecuencias que esto tiene en el desempeño y en la curva de aprendizaje. Nuevas plataformas como Django o Ruby on Rails, se centran precisamente en la reusabilidad del código relativo estos requisitos con bastante acierto, facilitando y focalizando el trabajo del desarrollador en lo funcional.

### **Generación de requerimientos**

La realización de un software capaz de contener de forma centralizada los antecedentes de cada alumno, permitirá a todos los interesados, obtener un reporte completo con datos detallados del alumno a consultar. Generando así un beneficio para la institución, pudiendo conocer en detalle la situación personal de cada alumno y creando instancias de resolución de problemáticas del alumno, atendiéndolo prontamente junto a su apoderado, especialmente en el registro de atrasos, inasistencias y evaluar sus causas. Así como también conocer el estado de salud y socioeconómico, para brindar apoyo o pesquisar situaciones de riesgo que vulneren la integridad física, emocional y ambiental, para realizar las intervenciones más apropiadas para cada caso.

### **Evaluación de requerimientos**

Se pudo constatar la inexistencia de sistemas que contengan información detallada y ordenada de los alumnos, pertenecientes al establecimiento educacional, como datos personales, registro de estado de salud, datos socioeconómicos, así como también registros de inasistencias, atrasos y sus causales. Lo que se traduce en tener que buscar la información requerida en múltiples archivadores, generando lentitud en los procesos y por ende aumentando el riesgo de extravío de dicha información.

### **Descripción de la solución.**

Elaboración de un entorno Web que contenga los datos personales, estado de salud y socioeconómicos del alumno, llevando un registro completo de inasistencias, atrasos y los motivos que lo ocasionan. Para que el establecimiento pueda realizar una intervención oportuna con el apoderado y el alumno.

### **Implementación y test**

Muy importante que continúe el proceso con la implementación y difusión del software, realizando las capacitaciones correspondientes en su utilización. Por lo que es necesario crear un programa de trabajo dentro del establecimiento, para optimizar los tiempos y que dentro de unos meses el software pueda ser implementado con todo lo requerido.

Adicional esto es bueno realizar las pruebas, al sistema ya sea rendimiento, de carga o pruebas a la base de datos, para conocer las anomalías que pueda presentar y así poder evitar algún riesgo de pérdida a futuro.

Cada uno de los formularios deben ser probados y validados con lo que la base de datos almacena, es necesario constar el hardware necesario donde se instalara el sistema, la habilitación de un servidor local en donde se almacenara la base de datos y el sistema en cuestión, y que cuente con un espacio físico previamente asignado.

Al tener claramente las actividades para realizar la gestión de requisitos, es mucho más fácil para el ingeniero de



software, ya que las tareas ya están bien definidas para su siguiente

Esto significa que la calidad del producto mejora tanto como sea posible y que se pueden manejar las solicitudes de para cambiar los requisitos de especificaciones.

Al mismo tiempo, se debe tener en cuenta la trazabilidad de los requisitos de gestión de proyectos.

Cabe señalar que la gestión de requisitos cambia a medida que avanza el proyecto y las actividades pueden variar.

Esto no significa que el trabajo realizado es sino solo que a través de las necesidades que surgen, el proceso debe adaptarse para lograr un buen sistema.

En ese sentido, el control de cambios aplicado y usado por el grupo desarrollador del proyecto ayuda a gestionar, coordinar, evaluar y decidir a los jefes de proyecto si los cambios solicitados son convenientes o no, siempre teniendo en cuenta qué es lo mejor para las partes interesadas y teniendo la precaución de no obstaculizar el cumplimiento de las metas y los tiempos de la realización del proyecto.

Aplicar un modelo de gestión de requisitos al proyecto durante el desarrollo es una técnica que utilizan los

<sup>i</sup> Material obtenido desde <https://bit.ly/3i5LAjs> - Ingeniería De Requerimientos Ingeniería De Software - monografias.com

<sup>ii</sup> <https://bit.ly/3AV1fXP> - Un modelo para la solución de requerimientos no alineados: El caso del Software lúdico para la divulgación Héctor G. Pérez-González Universidad Autónoma de San Luis Potosí [hectorgerardo@acm.org](mailto:hectorgerardo@acm.org)

<sup>iii</sup> <https://bit.ly/3wAz2IU> - Técnicas de levantamiento de requerimientos software.

<sup>iv</sup> <https://bit.ly/3yPOe8p> - Universidad de Granada - [elvex.ugr.es](http://elvex.ugr.es).

<sup>v</sup> <https://bit.ly/2U4dn8j> - [salonhogar.com/ciencias](http://salonhogar.com/ciencias)

<sup>vi</sup> <https://bit.ly/3xBZxJ3> - María Antonieta Abud Figueroa – Paper: Calidad en la Industria del Software. La Norma ISO-9126 - González Marcos, A. Alba Elías, F. y Ordieres Meré, J. (2014). Ingeniería de proyectos. Madrid, Spain: Dextra Editorial. Recuperado de <https://elibro.net/es/ereader/iacc/43933?page=106>.

<sup>viii</sup> González Marcos, A. Alba Elías, F. y Ordieres Meré, J. (2014). Ingeniería de proyectos. Madrid, Spain: Dextra Editorial.

ingenieros de desarrollo de software para analizar en detalle cada paso del desarrollo.

Estos modelos serán seleccionados gracias a un estudio que es el más práctico, según el proyecto sobre cómo circulan los datos en los sistemas, qué rama debe contener, cómo se divide, si esto incluye subdivisiones, entre otros.

El modelo es una guía para el ingeniero sobre cómo construir el sistema con la menor cantidad de errores posible.

Recuperado de <https://elibro.net/es/ereader/iacc/43933?page=107>.  
<sup>ix</sup> González Marcos, A. Alba Elías, F. y Ordieres Meré, J. (2014).

Ingeniería de proyectos. Madrid, Spain: Dextra Editorial. Recuperado de <https://elibro.net/es/ereader/iacc/43933?page=292>.

<sup>x</sup> Campderrich Falgueras, B. (2013). Ingeniería del software. Barcelona, Spain: Editorial UOC. Recuperado de <https://elibro.net/es/ereader/iacc/56294?page=16>.

<sup>xi</sup> <https://bit.ly/3iTnGoa> - Los 6 principios de la calidad del software.

<sup>xii</sup> <https://bit.ly/3rITiAX> - Requisitos para la Elicitación - [ecured.cu](http://ecured.cu).

<sup>xiii</sup> <https://bit.ly/3BFSNwh> - [scielo.conicyt.cl](http://scielo.conicyt.cl).

<sup>xiv</sup> <https://bit.ly/3kW6TDG> - Técnicas de elicitación de requerimientos.

<sup>xv</sup> <https://bit.ly/3kZEsVp> - Los 6 principios de la calidad del software.