



MÓDULO 3. CONCEPCIÓN DEL SISTEMA



[Imagen de Freepik](#)

DESCRIPCIÓN DEL MÓDULO 3

La concepción del sistema es el inicio para el desarrollo de un sistema. La concepción del sistema se ocupa de la génesis de una aplicación donde se tiene como propósito comprender el panorama general del negocio en base a las necesidades que presente.

El módulo inicia con la etapa de análisis. Se enfoca en la revisión de necesidades, que luego permiten el planteamiento de los requisitos partiendo de sus características y definiéndolos en requisitos funcionales y no funcionales. Así mismo se presentan los tipos de restricciones a tener en cuenta para la especificación de los requisitos.

Se continua con el documento visión y se finaliza con la matriz de trazabilidad. El documento visión permite registrar los puntos iniciales de la propuesta del negocio, sus generales y planteamiento de necesidades con la solución a abordar, permitiendo llevar un registro de los cambios y ajustes que se realicen. Por otro lado, la matriz de trazabilidad que plasma en un documento la correlación entre los requisitos con los casos de uso asociados a la solución que se desarrollan durante el proceso.





Contenido

Unidad 1. Análisis del problema	3
3.1. Análisis del problema	3
3.1.1. Definición del problema	3
Pasos del análisis de problemas	4
3.1.2. Descripción de stakeholders y usuarios	12
3.1.3. Necesidades.....	13
3.1.3.1. Derivación de las necesidades de usuario a partir del modelado del negocio.	13
3.1.4. Características.....	14
3.1.5. Requerimientos	15
3.1.5.1. Requerimientos funcionales	17
3.1.5.2. Requerimientos no funcionales	17
3.1.5.3. Restricciones de diseño	19
3.1.5.4. Aspectos de calidad de software.....	20
Unidad 2. Visión del sistema	22
2.1. Visión del sistema	22
Unidad 3. Elaboración de la matriz de trazabilidad	23
3.1. Elaboración de la Matriz de Trazabilidad del software.....	23
Conclusiones.....	25
Referencias	26



Unidad 1. Análisis del problema

Como la mayoría de los sistemas están diseñados para resolver un problema en particular, usaremos técnicas de análisis de problemas para asegurar que se entiende cuál es el problema.

3.1. Análisis del problema

Problemas y oportunidades están muy relacionados. El problema es en muchos casos una oportunidad, es decir una perspectiva del problema. El objetivo del análisis de problema es obtener un mejor entendimiento del contexto antes de comenzar el desarrollo del problema que se está resolviendo en el negocio.

3.1.1. Definición del problema

El análisis del problema es el proceso del entendimiento de necesidades de los usuarios y los problemas del mundo real, así como identificación de soluciones para esas necesidades encontradas.



Imagen tomada de [freepik](https://www.freepik.com)

El análisis de problemas es el proceso de comprender los problemas del mundo real y las necesidades del usuario y proponer soluciones para satisfacer esas necesidades [1].

Un problema se puede definir también como la diferencia entre como las cosas son percibidas y cómo son deseadas. Como solucionadores de problemas, se recomienda antes de decidir crear un nuevo sistema:

- Explorar alternativas simples de soluciones,
- Cambiar el deseo o la percepción del usuario, el cual puede ser el enfoque rentable al abordar un problema antes de saltar a una nueva solución de un sistema.



Pasos del análisis de problemas

El objetivo del análisis de problemas es lograr una **mejor comprensión**, antes de que comience el desarrollo del problema que se está resolviendo. Los pasos que deben tomarse para el análisis del problema de acuerdo con [2] se enumeran a continuación.

Pasos para análisis del problema:

1. Llegar a un acuerdo o consenso sobre la definición del problema.
2. Comprender las causas fundamentales: el problema detrás del problema.
3. Identificar las partes interesadas y los usuarios.
4. Definir el límite de solución del sistema.
5. Identificar las restricciones que se impondrán a la solución.

Descripción de pasos

1

Paso1. Lograr acuerdo sobre la definición del problema

El primer paso es llegar a un acuerdo sobre la definición del problema a resolver.

- Una de las maneras más sencillas de obtener este acuerdo es simplemente escribir el problema y ver si todos están de acuerdo, el enunciado del problema.
- A menudo es útil comprender algunos de los beneficios de una solución propuesta
- Hacer que el usuario describa los beneficios proporciona antecedentes contextuales adicionales sobre el problema. Al ver los beneficios desde el punto de vista del cliente, también se obtiene una mejor comprensión de la visión de las partes interesadas sobre el problema en sí.

El enunciado del problema

Puede resultarle útil escribir el problema en un formato estandarizado como el presentado en la tabla 1. Llenar la tabla para su aplicación es una simple técnica para ayudar a garantizar que todas las partes interesadas en el proyecto estén trabajando hacia el mismo objetivo.





Elemento	Descripción
El problema de ...	Describe el problema
Afecta a ...	Identifica stakeholder /usuarios afectados
Lo cual resulta en ...	Describe el impacto del problema en stakeholders y la actividad del negocio
Beneficios de la solución ...	Indica la solución propuesta y enumera algunos beneficios claves

Tabla 1. Formato de enunciado del problema.

2 Paso2. Comprender las causas fundamentales

Comprender las causas fundamentales es analizar el problema detrás del problema.

Una vez tenga una comprensión del problema más grande, puede usar una variedad de técnicas para comprender sus causas.

- Se recomienda analizar al problema conociendo las causas o la raíz que lo genera.
- Una de esas técnicas es el análisis de causa raíz, que es una forma sistemática de descubrir la causa raíz o subyacente de un problema identificado o un síntoma de un problema como el **diagrama de espina de pescado** que se muestra en la figura 1.

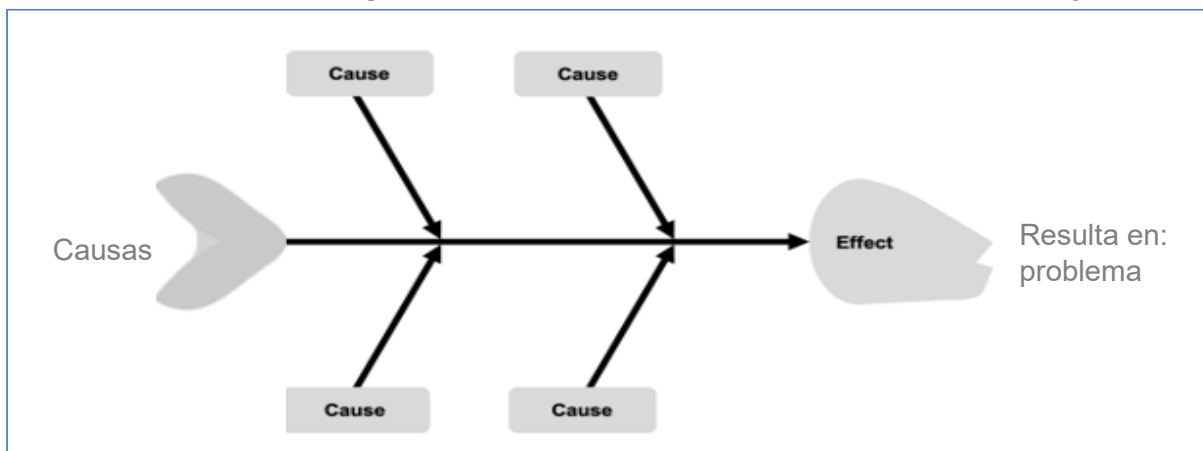


Figura 1. Diagrama de pescado



Ejemplo

- Considere una empresa ABC que fabrica y vende una variedad de artículos para el hogar y uso personal.
- A medida que la empresa aborda el *problema de la **rentabilidad** insuficiente*, utiliza técnicas de gestión de calidad total (TQM) para resolver problemas.
- Basándose en esta experiencia, la empresa se *concentró en su costo de incumplimiento*, que es el costo de todas las cosas que salen mal y producen desechos, chatarra y otros costos excesivos.
- Este costo incluye: reelaboración, desechos, insatisfacción del cliente, rotación de empleados y otros factores que son actividades de valor negativo.
- Se encontró que los desechos de producción, o "chatarra", son uno de los mayores contribuyentes después de la cuantificación de su costo de no calidad.

Este problema de exceso de chatarra, entonces, es el siguiente problema que la empresa está tratando de resolver, ya que afecta directamente el problema más grande del costo de la no conformidad, que a su vez afecta la rentabilidad.

- TQM indica el uso del **diagrama de espina de pescado** (ver Figura 1) para identificar los problemas detrás del problema. Cada fuente que contribuye a la falla se enumera como uno de los "huesos" en el diagrama, siendo en este caso las causas a representar como orígenes del problema

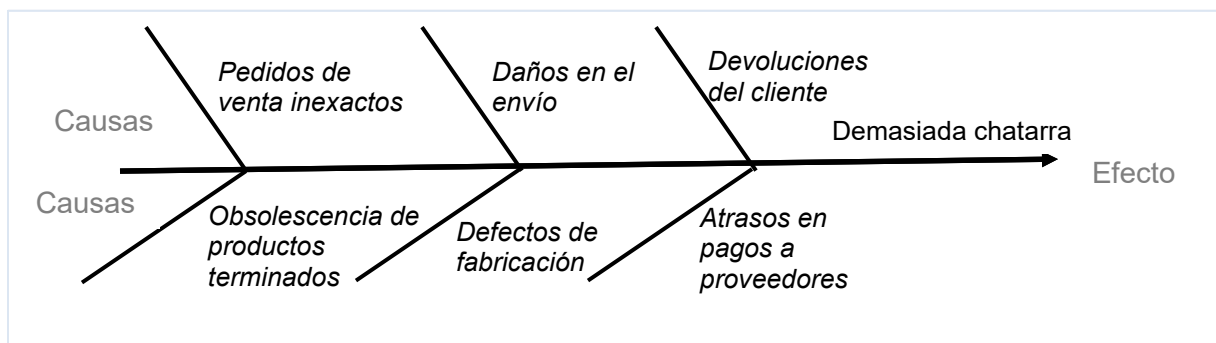


Figura 2. Diagrama de espina de pescado de las causas fundamentales



Abordar la causa principal de la raíz

Teniendo en cuenta las causas, se puede analizar cada una de ellas para identificar la raíz principal que origina el problema. Descubierto el problema de causa raíz, se analiza para dar soluciones.

Siguiendo con el ejemplo, se representan las causas que originan el efecto relacionado a rentabilidad resultando en el problema (ver figura 4)

Abordar la causa raíz, el problema a resolver se representa como sigue:

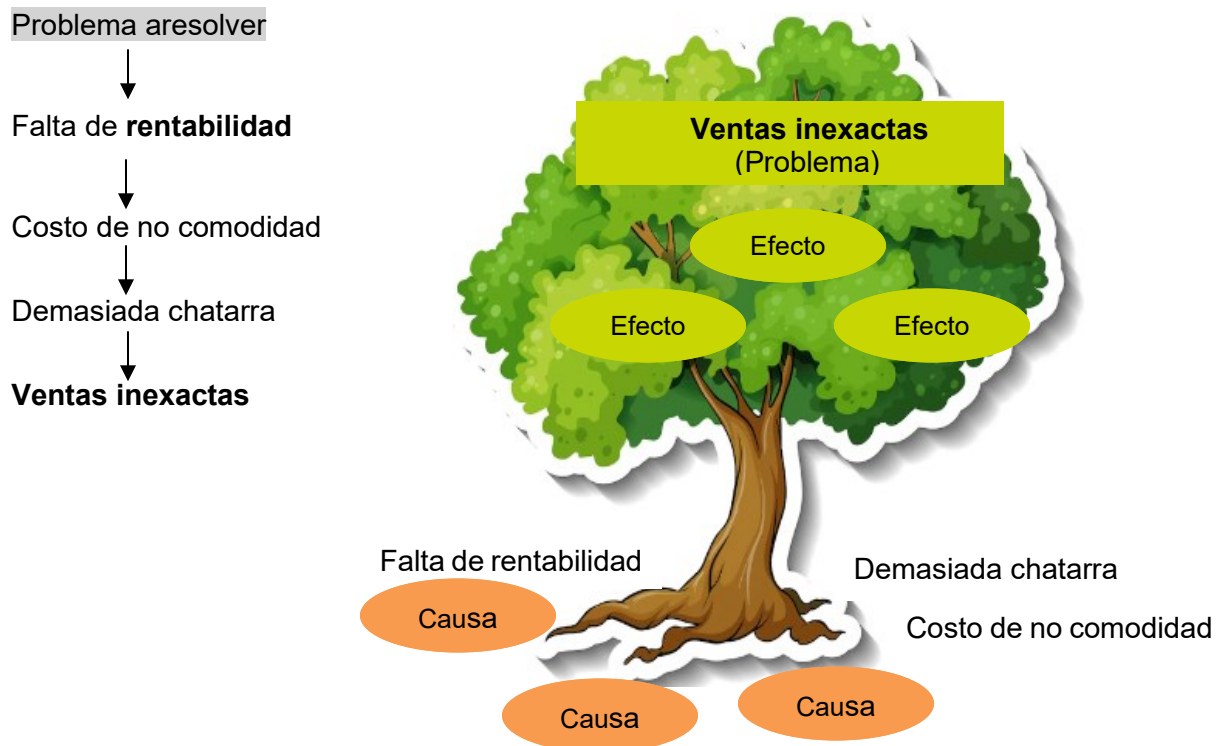


Figura 4. Representación de la raíz del problema

Una vez que hayamos identificado las órdenes de venta inexactas como la causa raíz de un problema que vale la pena resolver, podemos crear una declaración del problema para el problema de entrada de orden de venta, como se presenta en la tabla 2.



Enunciado del problema (ejemplo adaptado desde [2])

Elemento	Descripción
El problema de ...	(Descripción del problema) Ventas inexactas
Afecta a ...	(Identifica stakeholder /usuarios afectados) Personal de órdenes de venta, clientes, fabricante, servicio al cliente, entrega
Lo cual resulta en ...	(Describe el impacto del problema en stakeholders y la actividad del negocio) Aumento de desechos, costos de manejo excesivos, insatisfacción del cliente y disminución de la rentabilidad
Beneficios de una solución ...	Indica la solución propuesta y enumera algunos beneficios claves Crea un nuevo sistema para abordar el problema que incluye <ul style="list-style-type: none"> • Mayor precisión de las órdenes de venta en el punto de entrada. • Informes mejorados de los datos de ventas a la gerencia • Mayor rentabilidad

Tabla 2. Descripción de problema de ventas inexactas



3

Paso 3. Identifique los stakeholders y los usuarios

El término stakeholder fue acuñado por Edward Freeman quien lo definió como *todas aquellas personas o entidades que pueden afectar o son afectados por las actividades de una organización*. Como se define en [4] es una persona u organización que influye en los requisitos de un sistema o que se ve afectada por ese sistema.

Resolver un problema implica satisfacer las necesidades del diverso grupo involucrados. Un **stakeholder**, por definición, es cualquier grupo o individuo que puede afectar o se ve afectado por el logro de los objetivos de la organización [3].

- Stakeholders tienen perspectivas variadas del problema y necesidades varias direccionadas a la solución.
- Un *stakeholder* de acuerdo con [2] es cualquiera que podría ser afectado por la implementación de un nuevo sistema o aplicación.

Son las personas u organizaciones que se verán afectadas por el sistema y que tienen influencia directa o indirecta en los requisitos del sistema.

Las siguientes preguntas como menciona [2] pueden ayudar a identificar cuáles de los stakeholders tienen necesidades particulares:

¿Quiénes son los usuarios del sistema?

¿Quién es el cliente (comprador) para el sistema?

¿Qué otro puede ser afectado por las salidas que se producen en el sistema?

¿Existen otros usuarios internos o externos del sistema cuyas necesidades deben ser contempladas en la solución?

¿Quién mantendrá el sistema nuevo?, ¿Hay otro más?



Para el **ejemplo** del reemplazo del sistema órdenes de venta, el usuario primordial son los empleados que capturan los pedidos de entradas de las ventas. Ver tabla 3 para los usuarios y stakeholder involucrados.

Usuario	Stakeholder
Empleado encargado de la entrada de órdenes de ventas	Equipo desarrollador
Supervisor de orden de ventas	Jefe oficial de finanzas
Control de producción	Administrador de producción
Empleado encargado de facturar	Equipo desarrollador

Tabla 3. Identificando usuarios y stakeholder

Identificación de los stakeholders

La identificación de los stakeholders es una de las actividades iniciales en el análisis del problema. Primero se debe determinar quiénes son y qué importancia tienen. Para identificar los roles relevantes de las partes interesadas, se busca personas u organizaciones que [4]:

- Tienen un interés activo en el sistema porque realmente lo usarán o están directamente involucrados en los procesos que cambiará el sistema.
- Debe administrar, introducir, operar o mantener el sistema después de su implementación;
- Participan en el desarrollo del sistema como un arquitecto, desarrollador, evaluador, ingeniero de calidad o gerente de proyecto;
- Son responsables del negocio o proceso que el sistema soporta o automatiza;
- Tener un interés financiero (por ejemplo, ordenaron el sistema, lo pagaron o son responsables de su venta);
- Restringir el sistema como reguladores; o se ven afectados negativamente por el sistema (las llamadas partes interesadas negativas).



Importante: Como menciona [4], no todas las partes interesadas son igualmente importantes, por lo que se debe priorizar al actor identificado papeles, por ejemplo, en crítico, mayor y menor. Por lo general, basar la priorización en la evaluación del riesgo en el que se incurre al ignorar o descuidar una parte interesada.



4

Paso 4. Definir el límite de solución del sistema

Se debe entender el problema y las consideraciones de una solución potencial. Las fronteras o límites definen la solución entre el mundo real y lo que le rodea. Si se tuviera que construir o modificar una parte en particular del sistema, esa va ser parte de la solución y dentro de la frontera. Si no es externo al sistema. En ese sentido, se divide en dos partes: el sistema y las cosas con lo que va interactuar el sistema (ver figura 5).

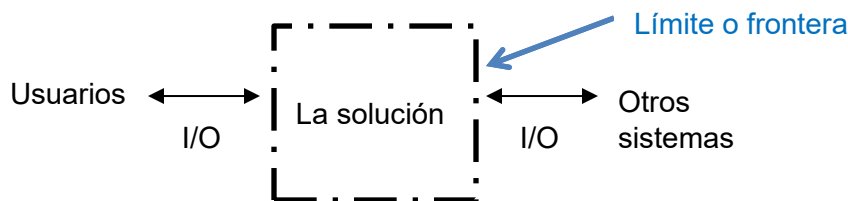


Figura 5. Representación de frontera según la solución en el negocio

5

Paso 5. Identificar las restricciones que se impondrán a la

Una **restricción** es un acuerdo de libertad que se tiene para una solución. Debido a que cada restricción tiene el potencial para limitar nuestras disponibilidades para la entrega de la solución, cada una de ellas debe ser considerada cuidadosamente como parte del proceso de planeación.

- Es responsabilidad de los que abordan la solución del problema comprender las posibles fuentes de restricciones para cada entorno de aplicación específico, así determinar el impacto de cada restricción en los posibles espacios de solución.





Siguiendo el ejemplo de sistema órdenes de venta, se presenta ejemplo de restricciones para este sistema en la tabla 4.

Fuente	Restricción
Sistema	El área ocupada por las aplicaciones en el servidor debe ser inferior a 20 MB.
Presupuesto de equipo	El sistema debe desarrollarse en el servidor y host existente; se puede proporcionar nuevo hardware de cliente para los usuarios.
Presupuesto de personal	Los recursos de personal son fijos; no es posible la subcontratación.
Operaciones	Una copia exacta de los datos del pedido de ventas debe permanecer en la base de datos heredada hasta por un año.

Tabla 4. Ejemplo de restricciones según la fuente de la misma en el negocio

3.1.2. Descripción de stakeholders y usuarios

Dentro de una solución hay afectados de alguna manera por la solución de un nuevo sistema.



- Una vez que se han identificado cada uno de los stakeholders y los usuarios, las necesidades que ellos tengan se procede a identificarlas.

Categorías de stakeholders:

- **Primario:** uso frecuente que interactuarán directamente. Usualmente llamados usuarios.
- **Secundario:** uso ocasional o a través de otra persona. Reciben la salida y proveen la entrada al sistema sin interactuar con el sistema. Indirectamente son usuarios del sistema
- **Terciario:** No se involucran con el sistema, pero se ven afectados por su éxito o fracaso
- **Facilitadores:** los que tienen a su cargo el desarrollo y la puesta en producción del sistema.





3.1.3. Necesidades

El equipo involucrado en el desarrollo de un sistema debe conocer las necesidades de todos los stakeholders involucrados. Pueden encontrar conflictos, pero se deben establecer prioridades, en donde los primarios son usualmente los de mayor prioridad son los primarios y dependiendo del caso pueden variar.

Identificar las necesidades de los usuarios provee un punto de partida para especificación de los requerimientos del sistema. Es decir, aquello que el sistema debe hacer y la funcionalidad que debe tener para apoyar en las tareas de los usuarios

3.1.3.1. Derivación de las necesidades de usuario a partir del modelado del negocio.

Las necesidades del usuario son las que centran en los requisitos de usuario, así satisfacen las necesidades del cliente.

- Los requisitos del usuario son declaraciones en lenguaje natural junto con los diagramas correspondientes (tablas, formularios, diagramas intuitivos) que detallan los servicios proporcionados por el sistema y las restricciones operativas que debe cumplir [5]. No contienen detalles del diseño del sistema y no utilizan expresiones técnicas o notaciones formales
- Estas necesidades de usuario, llamado requisitos del usuario generalmente se documentan en un Documento de requisitos del usuario (URD) utilizando texto narrativo.

Los requisitos del usuario, a menudo denominados necesidades del usuario, describen lo que el usuario hace con el sistema, como las actividades que los usuarios deben poder realizar. Estas necesidades de usuario generalmente son aprobados por el usuario y se utilizan como entrada principal para crear requisitos del sistema [6].



3.1.4. Características

Al identificar las necesidades hay que determinar las características a través de aspectos como:

- Qué es lo que están tratando de hacer/lograr los usuarios.
- Cómo lo están haciendo actualmente.
- Si ellos pudieran lograrlo más efectivamente, lo apoyaría con una alternativa diferente.

Por lo anterior, las **características** de las necesidades:

- Representan aquellos aspectos que describen lo relacionado con el problema
- Descripción detallada de la necesidad y como enfocarse para sus alternativas de **solución**.

En este contexto: ¿Qué debe hacer para afrontar y dar solución a una necesidad?

En la tabla 5 se presentan ejemplos de necesidades y cómo son las características de la solución.

Si la necesidad es:	La característica de la solución puede ser:
1. Que se pueda encontrar un libro específico sin escribir en un campo de texto	Se despliegue un listado de libros como los para que puedan ser seleccionados
2. Cliente desea saber qué libros están disponibles y cuáles no en la librería	Se despliegue en listado de libros disponible en depósito
3. Se requiere que ciertas funciones sólo las puedan realizar determinados usuarios	<ul style="list-style-type: none"> • Se controle el uso de las distintas funciones de acuerdo con las categorías de usuario. • Incorporación de control de acceso a ciertos usuarios

Tabla 5. Ejemplo de necesidades y sus características asociadas a solución

Las necesidades tienen asociadas procesos en el negocio. Por consiguiente, la identificación de necesidades guía la solución hacia los procesos que realiza el negocio. También conociendo los procesos, de forma inversa, se extraen y analizan las necesidades en cada proceso para orientar las características con la solución. En la siguiente tabla 6 se representan las necesidades, características de la solución asociadas a procesos en el negocio



Si la necesidad es:	La característica de la solución puede ser:	Proceso asociado en el negocio
1. Que se pueda encontrar un libro específico sin escribir en un campo de texto	Se despliegue un listado de libros como los para que puedan ser seleccionados	Consulta de libros por cliente
2. Cliente desea saber qué libros están disponibles y cuáles no en la librería	Se despliegue en listado de libros disponible en depósito	Consulta de libros en inventario
3. Se requiere que ciertas funciones sólo las puedan realizar determinados usuarios	<ul style="list-style-type: none"> • Se controle el uso de las distintas funciones de acuerdo con las categorías de usuario. • Incorporación de control de acceso a ciertos usuarios 	Actualización de inventario Registro de usuario /Inicio de sesión

Tabla 6. Necesidades, características de la solución y asociadas a procesos en el negocio

3.1.5. Requerimientos

Los requisitos o requerimientos para un sistema son las descripciones de los servicios que debe proporcionar un sistema y las restricciones sobre su funcionamiento. Estos requisitos como se indica [7] reflejan las necesidades de los clientes de un sistema que sirva para un determinado propósito en la organización, así, por ejemplo: controlar un dispositivo, realizar un pedido o encontrar información.

De acuerdo con el estándar IEEE 729, un requisito se define de la siguiente manera:

1. Una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. Una condición o capacidad que debe cumplir o poseer un sistema o componente del sistema para satisfacer un contrato, estándar, especificación u otros documentos impuestos formalmente.
3. Una representación documentada de una condición o capacidad como en 1 y 2.

Un requisito es una declaración sobre un producto esperado en donde se especifica lo que debe hacer o cómo debe realizar determinadas funciones según sea creado [8].





Clases de requisitos

Durante el proceso de identificación de requisitos se presentan diferentes niveles de descripción como se menciona en la tabla 7.

Requisitos de usuario	<p>Enunciados en lenguaje natural más diagramas de los servicios que brinda el sistema y sus restricciones operativas [9]. Escrito para el usuario.</p> <p><u>Ejemplo:</u></p> <p>Necesitamos poder revisar la ortografía de los documentos en varios idiomas.</p>
Requisitos del sistema	<p>Contiene la funcionalidad que necesita el sistema para cumplir con los requisitos del usuario estableciendo descripciones detalladas de los servicios del sistema [7] y [9]. Escrito como un contrato entre el cliente y el contratista.</p> <p><u>Ejemplo:</u></p> <p>El sistema debe poder revisar la ortografía de los documentos y proporcionar funciones de autocorrección. Habrá soporte para los siguientes idiomas, inglés, francés y alemán.</p>

Tabla 7. Clases de requisitos

Ejemplo de requisito descrito como **especificación de software** el cual va dirigido principalmente a los desarrolladores. Ejemplo adaptado de [9]

1. Verificación de resultados ortográfico (CheckResult) (palabra de cadena, diccionario del diccionario)

- La palabra se define en una cadena con formato UNICODE
- El CheckResult se define en 1 y contiene un indicador si la palabra se ha encontrado o no, además de un objeto Vector que contiene una lista de otras posibles sugerencias de palabras dependiendo de si la palabra se ha encontrado o no.
- El corrector ortográfico usará tablas Hashing para mejorar la eficiencia del código.





3.1.5.1. Requerimientos funcionales

Declaraciones de servicios que el sistema debe proporcionar, cómo debe reaccionar el sistema a entradas particulares y cómo debe comportarse el sistema en situaciones particulares [9].

- Estos requisitos describen las funciones que el software debe ejecutar.
- También pueden conocerse como características, capacidades del producto, funcionalidad o servicios del sistema [10].
- **Ejemplo:** Proporcionar un canal de comunicación para un usuario al realizar la compra.

3.1.5.2. Requerimientos no funcionales

Son requisitos sobre restricciones en los servicios o funciones que ofrece el sistema tales como restricciones de tiempo, restricciones en el desarrollo proceso, estándares y otros. Usualmente definido en el sistema como un todo [9]

- Los requisitos NO Funcionales actúan para hacer cumplir ciertas restricciones en la solución [10].
- Usualmente se describen como la versión ampliada de los requisitos del usuario que los ingenieros utilizan al comienzo del diseño de un sistema.
- Pueden clasificarse como requisitos de: calidad, usabilidad, tiempo de respuesta, confiabilidad, capacidad de recuperación, mantenibilidad, escalabilidad, rendimiento, seguridad, organizacional y otros.
- **Ejemplo:** El canal de comunicación con el usuario sea a través de correo electrónico indicado por el usuario durante los cinco primeros días de realizada la compra.

Nota: Los requisitos no funcionales que especifican todos los requisitos restantes no cubiertos por los requisitos funcionales pueden ser llamados también requisitos complementarios según sugerencia en [6]. Así evitar confusión con el término “No Funcional” ya que son significa que no trabajen o no sean funcionales. También, los requisitos complementarios a veces se denominan requisitos de calidad de servicio.

En la tabla 7 se muestra en resumen de lo que indican los requisitos funcionales y No funcionales de acuerdo con información descrita en [8].





Requisitos funcionales (RF)	Requisitos No funcionales (RNoF)
Dicen: lo que debe hacer el sistema	Dicen: qué restricciones hay en el sistema y su desarrollo para que funcione
Enfoca: en el “qué” debe hacer el sistema	Enfoca: en el “cómo” lo va a hacer el sistema
<p>Ejemplo 1: Para un procesador de textos:</p> <ul style="list-style-type: none"> ▪ “Debe poder admitir una variedad de estilos de formatos” 	<p>Ejemplo 1: Para un procesador de textos: <i>(Puede descomponerse en requisitos más específicos que detallen el tipo de formato requerido)</i></p> <p>“El formato debe admitir párrafo, por carácter y por documento, hasta un nivel muy específico, como que el formato de caracteres donde incluya 20 tipos de letra, cada uno con negrita, cursiva y opciones estándar</p>
<p>Ejemplo 2: Para un procesador de textos</p> <ul style="list-style-type: none"> ▪ “Debe ejecutarse en una variedad de plataformas de sistemas operativo” 	<p>Ejemplo 2: Para un procesador de textos</p> <ul style="list-style-type: none"> ▪ “Debe ser capaz de ejecutarse en una variedad de plataformas como: PC Mac, Windows y máquinas Unix”. <p>Otros RNoF:</p> <ul style="list-style-type: none"> ▪ “Debe poder funcionar en una computadora con 64 MB de RAM como mínimo”. ▪ “Debe entregarse en un plazo de seis meses”. (Esto representa una restricción en la actividad de desarrollo en sí misma más que en el producto que se está desarrollando).

Tabla 7. Resumen de requisitos funcionales y no funcionales.



3.1.5.3. Restricciones de diseño

Restricciones de diseño significa aquellas especificaciones y criterios de diseño contenidos en el contrato que especifican los estándares técnicos mínimos aceptables y definen los límites dentro de los cuales se desarrollará y llevará a cabo el diseño del Proyecto [11].

El diseño de interacción como indica requiere que se comprenda la funcionalidad requerida y las restricciones bajo las cuales el producto debe operar o desarrollarse. Para referirse a restricciones de diseño se presentan las categorías descritas por [8] que se mencionan.

1. Requisitos de datos

Capturan el tipo, la volatilidad, el tamaño, la cantidad, la persistencia, la precisión y el valor de las cantidades de los datos requeridos. Todos los dispositivos interactivos tienen que manejar cantidades mayores o menores de datos. Por ejemplo, si el sistema bajo consideración va a operar en el dominio de la aplicación de negociación de acciones, entonces los datos deben estar actualizados y ser precisos, y es probable que cambien muchas veces al día. Por ejemplo, en el dominio de la banca personal, los datos deben ser precisos, deben persistir durante muchos meses y probablemente años, son muy valiosos y es probable que sean muchos.

2. Medio ambiente

Se refieren a las circunstancias en las que se espera que funcione el producto en el contexto de uso. Se deben considerar cuatro aspectos del medio ambiente al establecer los requisitos.

Se deben considerar cuatro aspectos del medio ambiente al establecer los requisitos

- **Entorno físico**, como la cantidad de iluminación, ruido y polvo que se espera en el entorno operativo. ¿Necesitarán los usuarios usar ropa protectora, como grandes guantes o cascos, que podrían afectar la elección del paradigma de interacción?
- **Entorno social**, como la colaboración y la coordinación, deben explorarse en el contexto del desarrollo actual. Por ejemplo, ¿será necesario compartir los datos? Si es así, ¿el intercambio tiene ser sincrónico, por ejemplo, ¿todos deben ver los datos a la vez, o asincrónico, por ejemplo, dos personas que crean un informe se turnan para editarlo y agregarlo?





- **Organizacional**, qué tan bueno es el soporte al usuario, qué tan fácil se puede obtener y si hay instalaciones o recursos ¿para entrenamiento? ¿Qué tan eficiente o estable es la infraestructura de comunicaciones? ¿Qué tan jerárquica es la gestión?
- **Técnico**, por ejemplo, ¿con qué tecnologías se ejecutará el producto o necesitará ser compatible, y qué limitaciones tecnológicas podrían ser relevantes? Los requisitos del usuario capturan las características del usuario previsto.

3. Usabilidad.

Los requisitos con restricciones de usabilidad capturan los objetivos de usabilidad y las medidas asociadas para un producto en particular. Las medidas específicas para los objetivos de usabilidad del producto se establecen y acuerdan al principio del proceso de desarrollo y luego se revisan y se utilizan para realizar un seguimiento del progreso a medida que avanza el desarrollo. Esto asegura que la usabilidad tenga la debida prioridad y facilita el seguimiento del progreso.

Se describe una serie de objetivos de usabilidad: eficacia, eficiencia, seguridad, utilidad, capacidad de aprendizaje y memorabilidad.

3.1.5.4. Aspectos de calidad de software

Calidad en el diseño de software según indica Pressman incluye el grado en el que el diseño *cumple las funciones y características especificadas en el modelo de requerimientos*. La calidad de la conformidad se centra en el grado en el que la implementación se apega al diseño y en el que el sistema resultante cumple sus metas de requerimientos y desempeño

satisfacción del usuario =
producto que funciona + buena calidad + entrega dentro del presupuesto y plazo

También se destaca la norma ISO/IEC 9126 [12] que distingue 4 tipos de niveles de calidad: calidad en uso, calidad externa, calidad interna y calidad de proceso. Clasifica los atributos de calidad del software en seis características: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad, que se subdividen en subcaracterísticas como se presenta en la figura 6. Las subcaracterísticas se pueden medir mediante métricas internas o externas



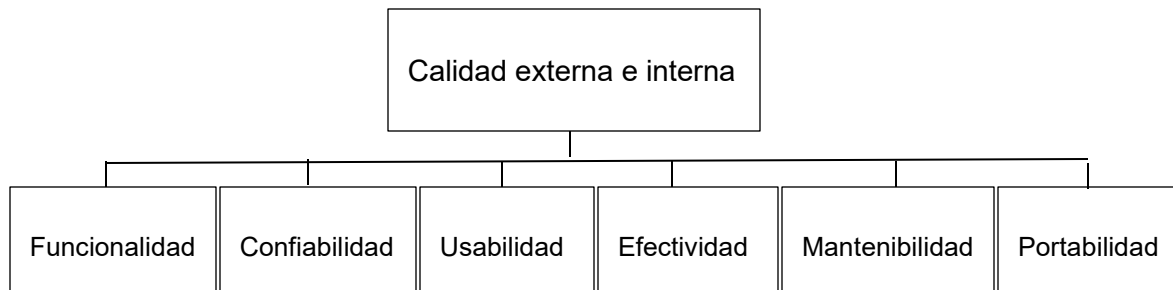


Figura 6. Atributos para los requisitos de restricciones de calidad.

Funcionalidad: precisión de idoneidad, seguridad de interoperabilidad

Confiabilidad: madurez tolerancia a fallos recuperabilidad

Usabilidad: comprensibilidad, capacidad de aprendizaje, operabilidad, atractivo

Efectividad: comportamiento del tiempo, utilización de recursos

Mantenibilidad: analizabilidad, cambiabilidad, comprobabilidad

Portabilidad: adaptabilidad, instalabilidad, coexistencia, reemplazabilidad

FURPS, el cual es un acrónimo que representa un modelo para clasificar atributos de calidad de software o requisitos no funcionales, desarrollado en Hewlett-Packard, y luego se agregó otros atributos para enfatizar varios atributos de calidad [13]. Los atributos de calidad según FURPS se mencionan en los siguientes puntos.

- Funcionalidad: conjunto de características, capacidades, generalidad, seguridad
- Usabilidad: Factores humanos, Estética, Consistencia, Documentación
- Confiabilidad: Frecuencia/gravedad de la falla, Recuperabilidad, Previsibilidad, Precisión, tiempo medio hasta el fallo
- **Rendimiento**: velocidad, eficiencia, consumo de recursos, rendimiento, tiempo de respuesta
- **Capacidad de soporte**: capacidad de prueba, extensibilidad, adaptabilidad, capacidad de mantenimiento, compatibilidad, capacidad de configuración, capacidad de servicio, capacidad de instalación, capacidad de localización, portabilidad



Unidad 2. Visión del sistema

En este punto se menciona lo relacionado al documento visión que se realiza en el curso.

2.1. Visión del sistema

El documento visión es un documento que contiene la descripción de la propuesta de proyecto. Incluye el conjunto de necesidades que dan lugar a la iniciación del proyecto y sirven para definir lo que va a desarrollarse en el proyecto.

Descripción

Este documento contiene información sobre la oportunidad del negocio, descripciones del producto, supuestos y restricciones, entre otros. Este documento permite:

- Poder plasmar cual es el resultado global que se quiere lograr en el proyecto.
- Lograr un control de cambios que se realicen a lo largo del proyecto.
- Que sirva de soporte a la especificación de las características software y de los atributos de estas.
- Ser una consulta permanente por parte del grupo que desarrolla el proyecto y del cliente para no perder el rumbo

Estructura del documento visión

Incluye el objetivo, el alcance, las definiciones, los acrónimos, las abreviaturas, las referencias y una **visión** general de todo el **documento**. Algunos de los puntos que lo componen son los siguientes.

1. Oportunidad de negocio.
2. Declaración de oportunidad.
3. Análisis de beneficios.
4. Concepto de Solución.
5. Requerimientos.
6. Alcance.
7. Estrategia de entrega de versiones.
8. Criterios de aceptación.



Ventajas del documento visión

Su importancia radica en que se utiliza, además del contenido de la descripción del proyecto, para poder "controlar" el cambio, a medida que avanza el proyecto. Por este motivo, va a ser requerido a cada grupo de trabajo en futuras entregas, que se estará revisando según se indique. No olvidar que tanto el cliente como el grupo deben tener la misma visión, por lo que el Grupo debe validar dicho documento con el Cliente.

Unidad 3. Elaboración de la matriz de trazabilidad

Para dar seguimiento a los cambios que se realicen al proyecto, se debe poder rastrear cada componente del sistema hasta el requisito original que causó su presencia. Este seguimiento se realiza a través de la matriz de trazabilidad

3.1. Elaboración de la Matriz de Trazabilidad del software

La trazabilidad es la medida en la cual se puede establecer una relación entre dos productos o más del proceso de desarrollo [14]. La matriz de trazabilidad se desarrolla en un documento para realizar un seguimiento de los requisitos y comprobar que se cumplen los requisitos actuales del proyecto. De allí que también suele llamarse matriz de seguimiento de trazabilidad. La trazabilidad de requisitos se refiere a la "habilidad de seguir el ciclo de un requerimiento" [14] desde que inicia a medida que avanzan (adelante y hacia atrás).

La matriz de trazabilidad de requisitos (RTM) es un documento que mapea y rastrea los requisitos del usuario con casos de prueba. Captura todos los requisitos propuestos por el cliente y la trazabilidad de los requisitos en un solo documento, entregado al final del ciclo de vida del desarrollo del Software [16].

Una matriz de trazabilidad se utiliza principalmente para [15]:

- Mantener al equipo al tanto de los requisitos que se le dan respuesta
- Asegurar el cumplimiento de los requisitos del proyecto.
- Una matriz de trazabilidad hacia atrás ayuda a ver qué casos de prueba se mapean con qué requisitos. Esto permitirá identificar si hay casos de prueba que no se relacionan con ningún elemento de cobertura. Si el caso de prueba no se relaciona con un





elemento de cobertura, no es necesario y debe eliminarse, o bien debe agregarse una especificación como uno o dos requisitos [17].

El seguimiento de los requisitos del proyecto puede ser un desafío, según su entorno de desarrollo. Algunos proyectos pueden tener solo unos pocos requisitos, mientras que otros pueden tener docenas como se presenta en la figura 7, la cual tiene asociado los requisitos con los casos de uso.

	<i>C1</i>	<i>C2</i>	<i>...</i>	<i>...</i>	<i>Cm</i>
<i>req1</i>		x			
<i>req2</i>	x				
<i>...</i>				x	
<i>...</i>					x
<i>reqn</i>		x	x		

Figura 7. Modelo de matriz de trazabilidad

En la figura 8 se muestra otro formato de matriz que incluye el número de requisito, la descripción de los requisitos y los casos de usos, los cuales. Puede agregar más columnas y hacerlo más efectivo como: prueba del caso de uso, estatus,

ID Requisito	Descripción de requisito	Solución	Nombre del caso de uso
	El usuario debería poder hacer ...		CU1, CU2,..., CUn
	El usuario debería poder hacer eso.		CU, CU2
	Al acceder, el usuario debería. .		CU1

Figura 8. Modelo de matriz de trazabilidad con



Conclusiones

En este módulo se presentaron diferentes puntos a tomar en cuenta en este módulo relacionados con la concepción del sistema. La concepción del sistema se encarga de la génesis de una aplicación. Inicialmente, se busca comprender tanto las necesidades negocio como la tecnología, para idear posibles soluciones.

Se puede utilizar una variedad de técnicas para entender el problema real y sus causas. Una de estas técnicas es análisis que causa la raíz que es una manera de descubrir la raíz que origina un problema identificado. Una manera de determinar las causas es preguntando directamente a las personas involucradas lo que ellos piensan lo origina. Dependiendo que tan serio sea el problema, puede ser necesario realizar una investigación más detallada.

Resolver un problema implica satisfacer las necesidades del diverso grupo de usuarios que están relacionados con la solución entre ellos los stakeholders. Un stakeholder es cualquiera que podría ser afectado materialmente por la implementación de un nuevo sistema o aplicación.

Luego que se realiza el análisis de las necesidades, serán los insumos para describir los requisitos que se trabajarán en el sistema. Los requisitos son de tipo funcional y no funcional. Los funcionales. Los funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Al hablarse de entradas, no necesariamente son sólo de las entradas de los usuarios, ya que pueden ser interacciones con otros sistemas, respuestas automáticas, procesos predefinidos [18], es decir “lo que hace el sistema”. Los no funcionales no se refieren directamente a las funciones específicas suministradas por el sistema, sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad y otros, en otras palabras “cómo lo hace el sistema” tomando en cuenta restricciones del sistema.

Se mencionó la importancia del contenido de la descripción del proyecto, para poder "controlar" el cambio, a medida que avanza el proyecto en el documento visión. Así mismo el seguimiento de desarrollo de los requisitos que se desarrollan en la matriz de trabilidad





Referencias

- [1] E. A. Javed, *Edu.pk*. [Online]. Available: http://web.uettaxila.edu.pk/CMS/SP2013/seSREbs/notes%5CSRE%20Lec_3.pdf. [Accessed: 29-Jul-2022].
- [2] D. Leffingwell - Managing Software Requirements: A Use Case Approach [PDF] [2u9vrk6qhu40], Vdoc.pub. [Online]. Available: <https://vdoc.pub/download/managing-software-requirements-a-use-case-approach-2u9vrk6qhu40>. [Accessed: 30-Jul-2022].
- [3] H. Sharp, A. Finkelstein, and G. Galal, "Stakeholder identification in the requirements engineering process," in *Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99*, 1999, pp. 387–391.
- [4] M. Glinz and R. J. Wieringa, "Stakeholders in Requirements Engineering," Uzh.ch. [Online]. Available: <https://www.ifi.uzh.ch/dam/jcr:3fd4da15-79bf-4870-90d0-f251a7d1a1a8/GlinzWieringa2007.pdf>. [Accessed: 31-Jul-2022].
- [5] GuidingCode, "What are user and system requirements in software engineering," GuidingCode, 27-Jan-2022. [Online]. Available: <https://www.guidingcode.com/user-and-system-requirements-in-software-engineering/>. [Accessed: 01-Aug-2022].
- [6] J. Parker, "Business, user, and system requirements," Enfocus Solutions Inc, 18-Feb-2012. [Online]. Available: <https://enfocussolutions.com/business-user-and-system-requirements/>. [Accessed: 01-Aug-2022].
- [7] Collegenote, "Concept of User and system requirements," Collegenote.net. [Online]. Available: <https://www.collegenote.net/curriculum/software-engineering-csit/54/309/>. [Accessed: 01-Aug-2022].
- [8] Y. Rogers, H. Sharp, and J. Preece, "INTERACTION' DESIGN beyond human-computer interaction," Cornell.edu, 2022. [Online]. Available: <https://arl.human.cornell.edu/879Readings/Interaction%20Design%20-%20Beyond%20Human-Computer%20Interaction.pdf>. [Accessed: 01-Aug-2022].
- [9] S. Coope, "Lecture 4 -Software Requirements," Liv.ac.uk. [Online]. Available: https://cgi.csc.liv.ac.uk/~coop/es/comp201/handouts/SE_L4.pdf. [Accessed: 01-Aug-2022].
- [10] G. Byrne, "Why you need to understand Software Requirements as a software engineer," freecodecamp.org, 27-Apr-2020. [Online]. Available: <https://www.freecodecamp.org/news/why-understanding-software-requirements-matter-to-you-as-a-software-engineer/>. [Accessed: 01-Aug-2022].
- [11] "Design requirements definition: 112 samples," Law Insider. [Online]. Available: <https://www.lawinsider.com/dictionary/design-requirements>. [Accessed: 02-Aug-2022].
- [12] Edu.au. [Online]. Available: <https://www.cse.unsw.edu.au/~cs3710/PMmaterials/Resources/9126-1%20Standard.pdf>. [Accessed: 02-Aug-2022].
- [13] L. Chung and J. C. S. do Prado Leite, "On Non-Functional Requirements in Software Engineering," in *Conceptual Modeling: Foundations and Applications*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 363–379.





- [14] B. Quito, "Trazabilidad En Proyectos De Software," Slideshare.net. [Online]. Available: <https://pt.slideshare.net/barcampquito/trazabilidad-en-proyectos-de-software/3>. [Accessed: 03-Aug-2022].
- [15] Simplilearn, "Requirements traceability matrix (RTM): Templates, types and tools for 2022," Simplilearn.com, 05-Dec-2019. [Online]. Available: <https://www.simplilearn.com/project-management-and-the-requirements-traceability-matrix-article>. [Accessed: 03-Aug-2022].
- [16] T. Hamilton, "What is requirements Traceability Matrix (RTM)? Example template," Guru99, 20-Jan-2020. [Online]. Available: <https://www.guru99.com/traceability-matrix.html>. [Accessed: 04-Aug-2022].
- [17] H. Sachdeva, "What to include in a software testing traceability matrix," Optimus Information Inc, 01-Feb-2013. [Online]. Available: <https://www.optimusinfo.com/blog/software-testing-traceability-matrix>. [Accessed: 04-Aug-2022]
- [18] R. Blog, "Requerimientos Funcionales y No Funcionales, ejemplos y tips," Medium, 20-Apr-2018. [Online]. Available: <https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a>. [Accessed: 04-Aug-2022].