

Redacción Avanzada

A close-up photograph of a person's hands writing in a notebook with a black pen. The hands are in focus, while the background is blurred, showing a white shirt and a keyboard, suggesting a professional or academic setting.



Bienvenida



Bienvenido(a) a la asignatura *Redacción Avanzada*, con la cual podrás comprender cómo deben redactarse los requerimientos de los sistemas de información que desarrollarás. En general, identificarás los métodos de elicitación de requerimientos, así como sus tipos y gestión.





Libros recomendados

- Baker, R. (2017). *Agile UX Storytelling: Crafting Stories for Better Software Development*. Apress.
- Patton, J. (2016). *User Story Mapping: Discover the Whole Story, Build the Right Product*. O'Reilly.



Unidad 1

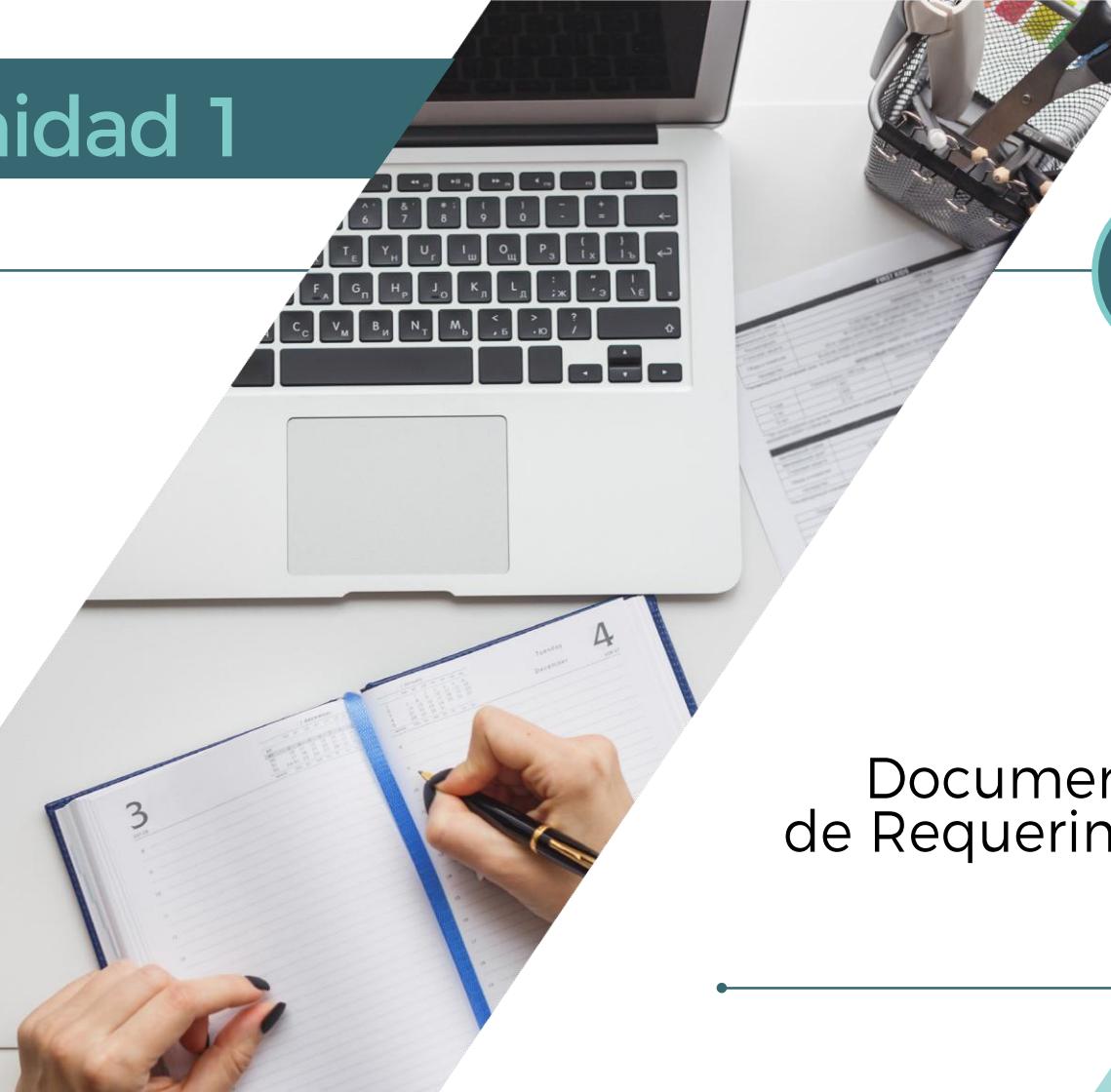
Redacción de Requerimientos de Software



Temario Unidad 1

1.1

Elicitación de
Requerimientos



1.2

Documentación
de Requerimientos



Introducción

La asignatura *Redacción Avanzada* tiene como objetivo el apoyarte a mejorar tus competencias de comunicación. Particularmente en la transmisión y recepción de documentos de software.

En esta primera unidad aprenderás a obtener los requerimientos mediante el proceso de elicitación, para posteriormente plasmarlos mediante el proceso de documentación.

Competencias a desarrollar



El alumno será capaz de eliciar requerimientos de software.



El alumno será capaz de redactar requerimientos de software.





Elicitación de Requerimientos

Involucrar al Cliente en el Desarrollo del Producto

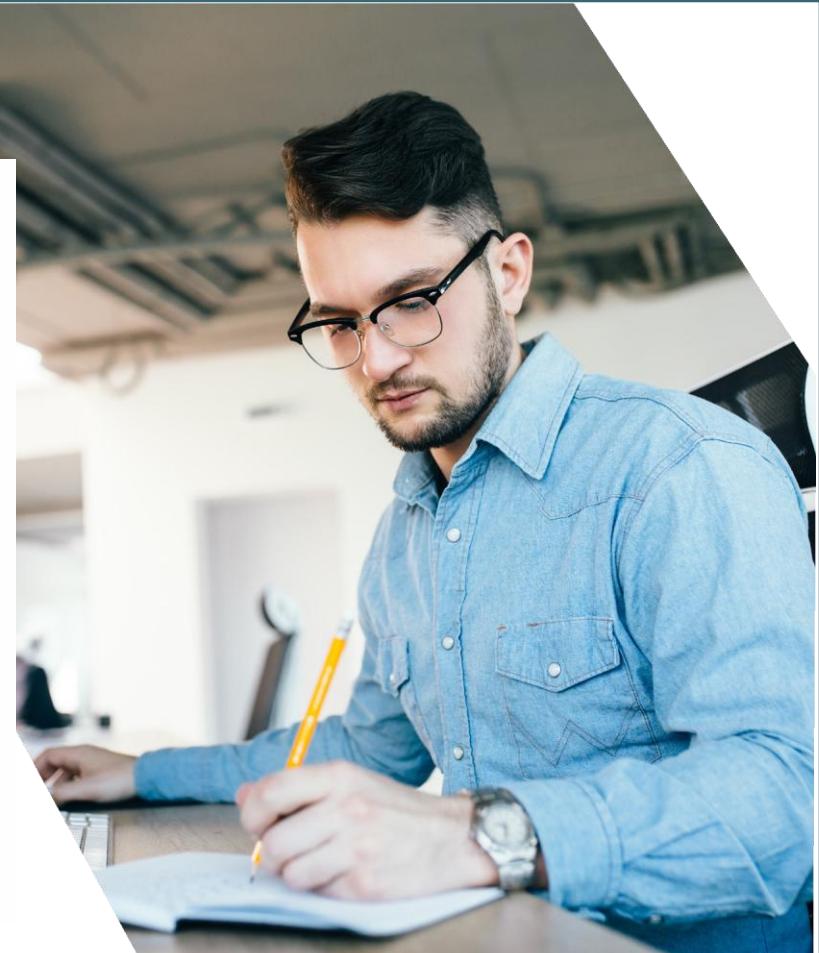
Aunque esto parece algo obvio, no siempre ocurre. La principal razón para esta omisión es el **hueco de expectativas** que se puede crear si no se realiza esta práctica.

Este hueco es la diferencia que existe entre lo que el cliente espera del producto y aquello que se le entrega. En consecuencia, es muy importante darle seguimiento porque, a pesar de que los requerimientos sean claros, siempre existe la posibilidad de que el cliente espere otra cosa.

Dar un seguimiento a estas expectativas es complicado, ya que se puede crear un hueco en cada interacción social.

En esta interacción social se involucran:

- Miembros del equipo de desarrollo de software,
- Miembros de la empresa de desarrollo que están asociados al proyecto y
- Miembros asociados al proyecto de la empresa cliente.





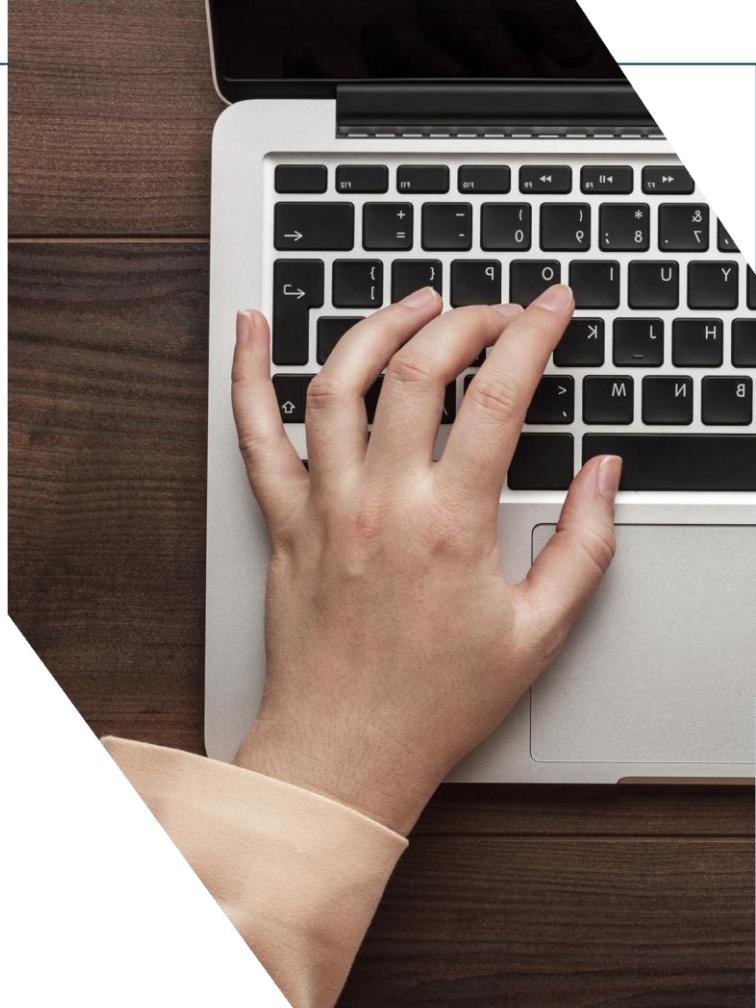
Incluso si la producción del software se realiza exactamente como el cliente lo solicita, existe la posibilidad de que el **producto final** no sea el adecuado.

El **cliente**, quien es el que paga y por tanto al que se le pregunta qué desea, funge un rol; pero el **usuario final del producto** desempeña otro papel.

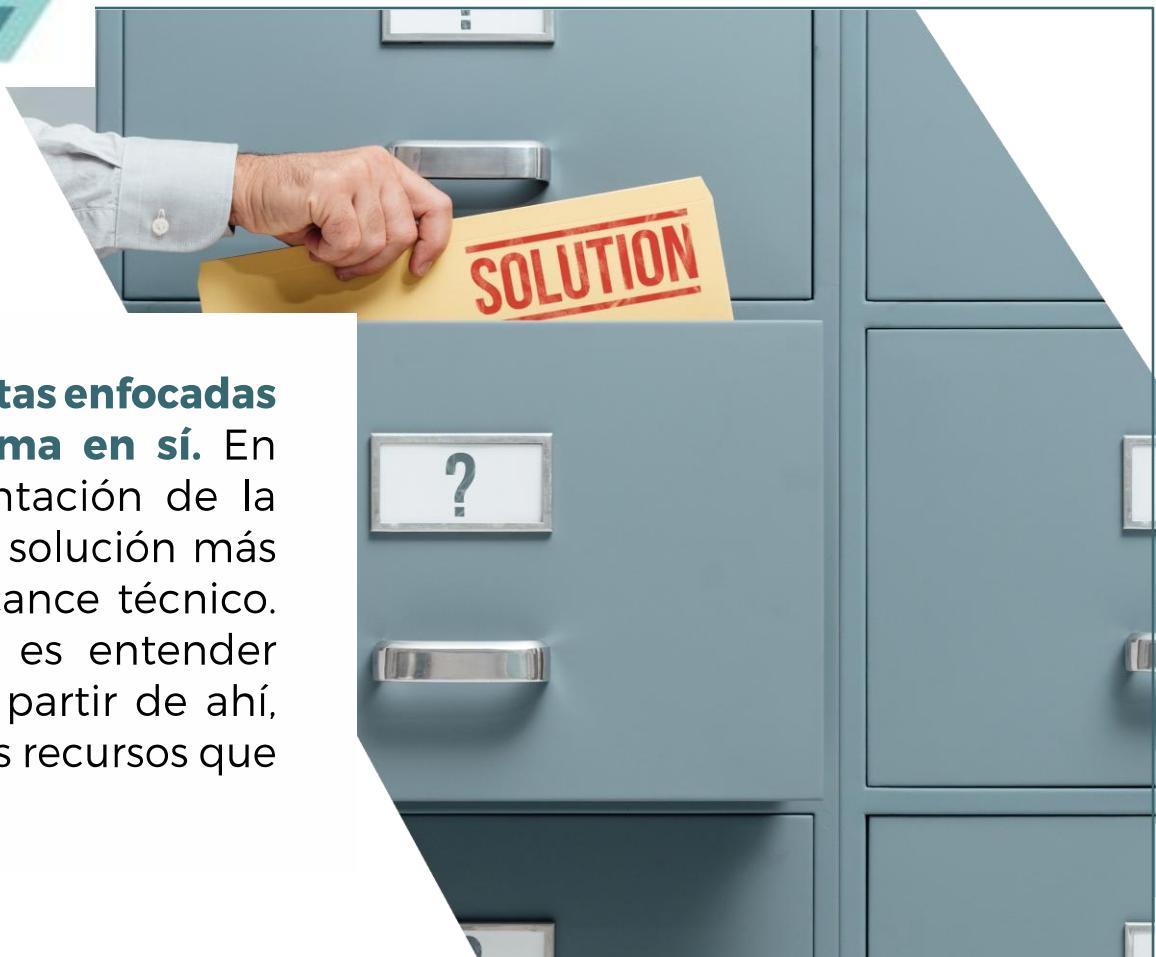
El usuario final puede tener otras expectativas distintas al cliente, ya que ambos tienen sus propias perspectivas del mismo producto.

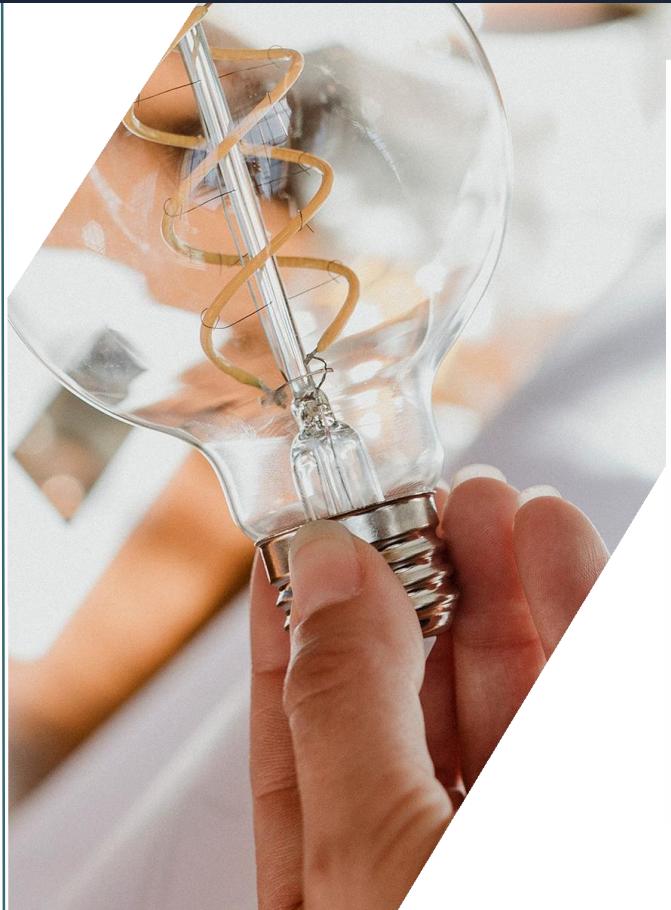
Por tanto, es importante que las personas que realicen la elicitation de requerimientos se comprometan con lo siguiente:

1. Hablar con el cliente y/o el usuario en los mismos términos de su negocio. Ni el cliente ni el usuario saben necesariamente de sistemas o cómo diseñarlos. Por tanto, no tienen la responsabilidad de hacer peticiones concretas. Es trabajo de quién esté realizando la elicitation identificar los problemas de negocios para poder proponer soluciones.



2. Ofrecer soluciones concretas enfocadas al negocio y no al sistema en sí. En muchas ocasiones es una tentación de la persona de sistemas ofrecer la solución más sencilla en términos de su alcance técnico. Sin embargo, ser profesional es entender la necesidad del negocio y, a partir de ahí, ofrecer la mejor solución con los recursos que se tengan.





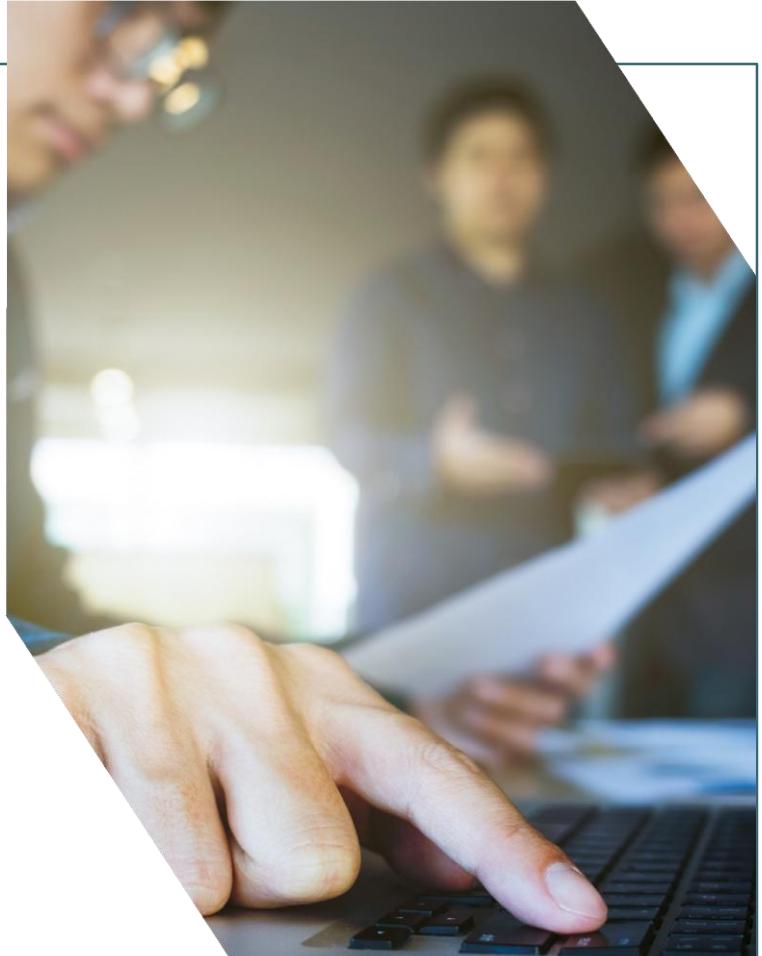
3. Ofrecer soluciones de negocio de calidad. Existen distintas definiciones de calidad, pero a grandes rasgos, como premisas básicas, se debe considerar:

- a)** Productos fáciles de usar,
- b)** Escalables,
- c)** Reusables y
- d)** Funcionales.

Además, parte de ofrecer una solución de calidad es **ser realista con los costos, alcances y tiempos**. Una pésima práctica es prometer cosas que se sabe de antemano que no se cumplirán en tiempo, forma o recursos.

Entre las **principales prácticas de elicitation de requerimientos** se encuentran las siguientes:

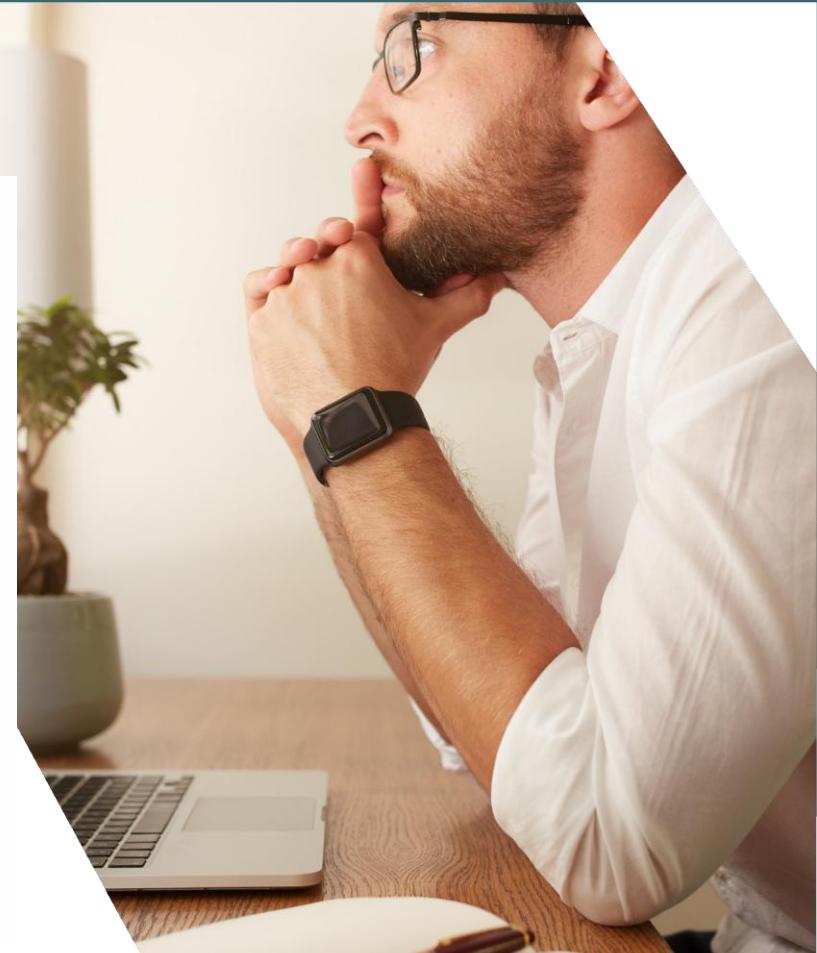
- **Definir la visión del producto y su alcance.** Contiene los requerimientos de negocio del producto. El alcance define qué estará y qué no estará en el producto.
- **Definir los usuarios.** Se deben describir todos los posibles usuarios del producto. Así como su frecuencia, privilegios, experiencia, actividades, actitudes y entorno laboral.





- **Seleccionar un responsable por tipo de usuario.** Cada tipo de usuario debe tener una persona que se responsabilice por definir los requerimientos y evaluar los productos de software enfocados a su rol.
- **Conducir grupos de enfoque o entrevistas con usuarios promedio.** Se debe promover en la medida de lo posible la reunión con los usuarios, donde se presenten los productos de software asociados a ellos, así como recuperar sus opiniones y experiencias.

- **Evaluar la mejor manera de representar el conocimiento.** Los requerimientos de los usuarios se pueden representar a través de casos de uso, historias de usuario o escenarios.
- **Identificar los eventos de sistema y sus respuestas.** Realizar una lista de los eventos externos al sistema y redactar cómo debe comportarse este ante ellos. Estos eventos pueden ser una señal de hardware, un lapso temporal o un disparador por software.



Elicitación de Requerimientos

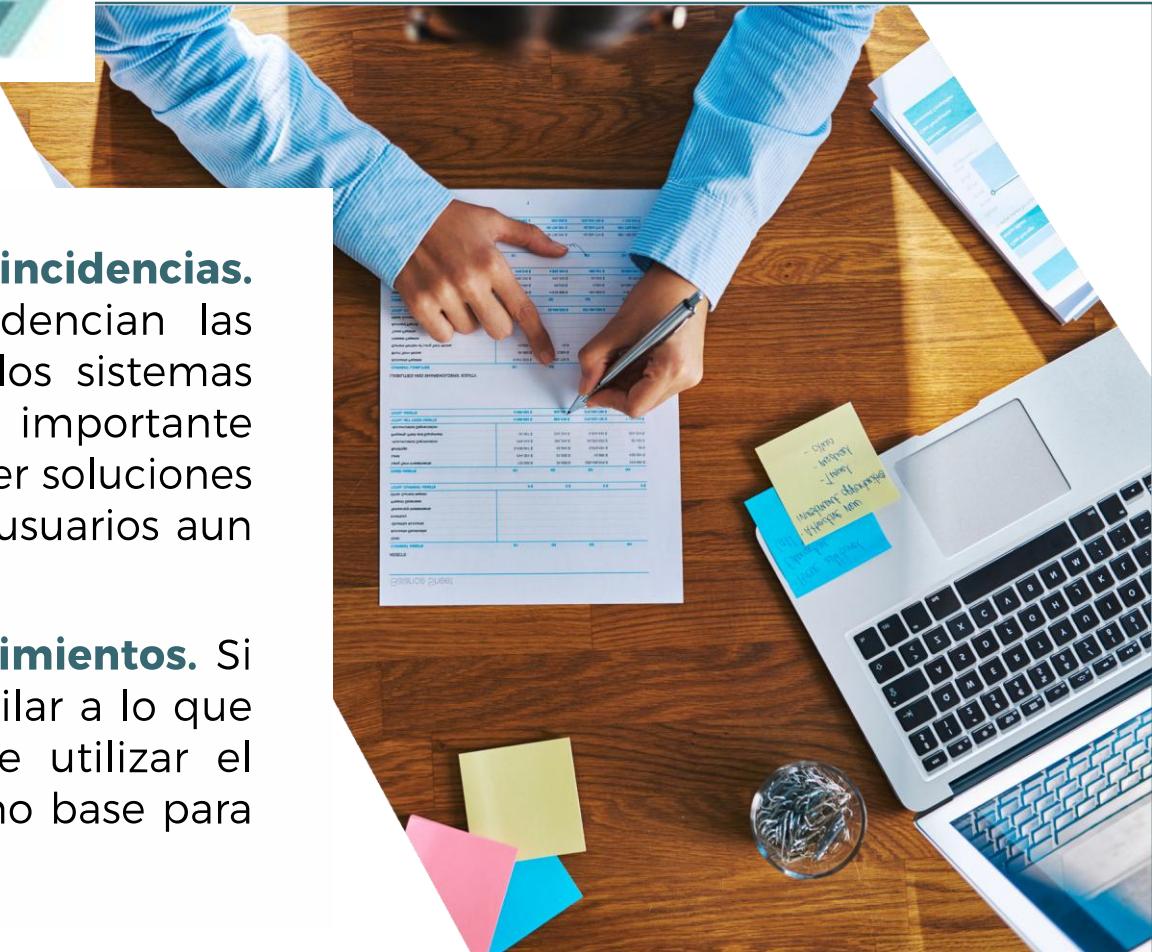


- **Observar a las personas realizar su trabajo.** Permite establecer el contexto del uso potencial de un nuevo producto de software.
- **Distribuir cuestionarios.** Esta es una manera relativamente económica de obtener datos desde una vastedad de usuarios, con poco tiempo y esfuerzo.
- **Realizar una revisión documental.** La revisión de documentos actuales puede revelar cómo funcionan realmente los procesos en una empresa.

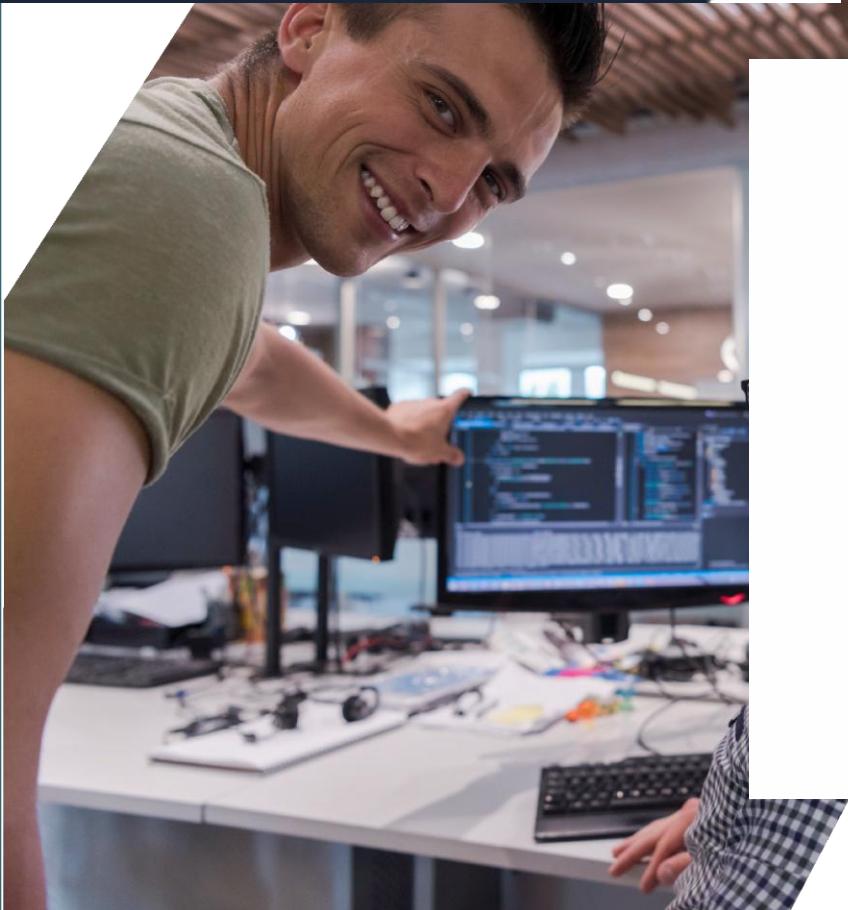
- **Revisar los reportes de incidencias.**

Este tipo de reportes evidencian las áreas de oportunidad de los sistemas y los procesos. Por ello, es importante analizarlos para poder ofrecer soluciones de negocio a los clientes y usuarios aun cuando no los pidan.

- **Volver a usar los requerimientos.** Si un usuario solicita algo similar a lo que otro haya pedido, se debe utilizar el requerimiento anterior como base para construir a partir de ahí.



Elicitación de Requerimientos



- **Modelar el entorno del producto de software.** Un diagrama de contexto evalúa cómo se insertará el nuevo producto dentro de una familia de productos que ya existe. Esto permite visualizar oportunidades y riesgos, ya que rara vez hay alguien que pueda observar todas las posibilidades de una modificación a los sistemas.
- **Crear interfaces y prototipos.** Cuando hay dudas o no se llega a acuerdos, lo ideal es construir versiones iniciales que permitan clarificar opciones.

- **Priorizar los requerimientos.** Con base en cada prioridad, se debe determinar cuál requerimiento será entregado en cada liberación o incremento. Estas prioridades deben revisarse constantemente conforme el proyecto evoluciona a lo largo del tiempo.
- **Crear el diccionario de datos.** Es importante crear un diccionario de datos, términos y estructura asociado al proyecto, de manera que todos los involucrados estén siempre conscientes de lo que está ocurriendo.





Video

Te invitamos a ver el
siguiente video:





Documentación de Requerimientos

Como hemos visto, la elicitación de requerimientos no solo se trata de preguntar al cliente qué desea. Por esta razón, comportarse como un profesional en el área involucra tomar la iniciativa y buscar cada posibilidad de mejora o error, así como su prevención.

Cuando las cosas van bien, se puede pensar que la documentación es una pérdida de tiempo. Sin embargo, cuando las cosas no salen como se planean, **la documentación puede ayudar a clarificar situaciones** y corregir errores.

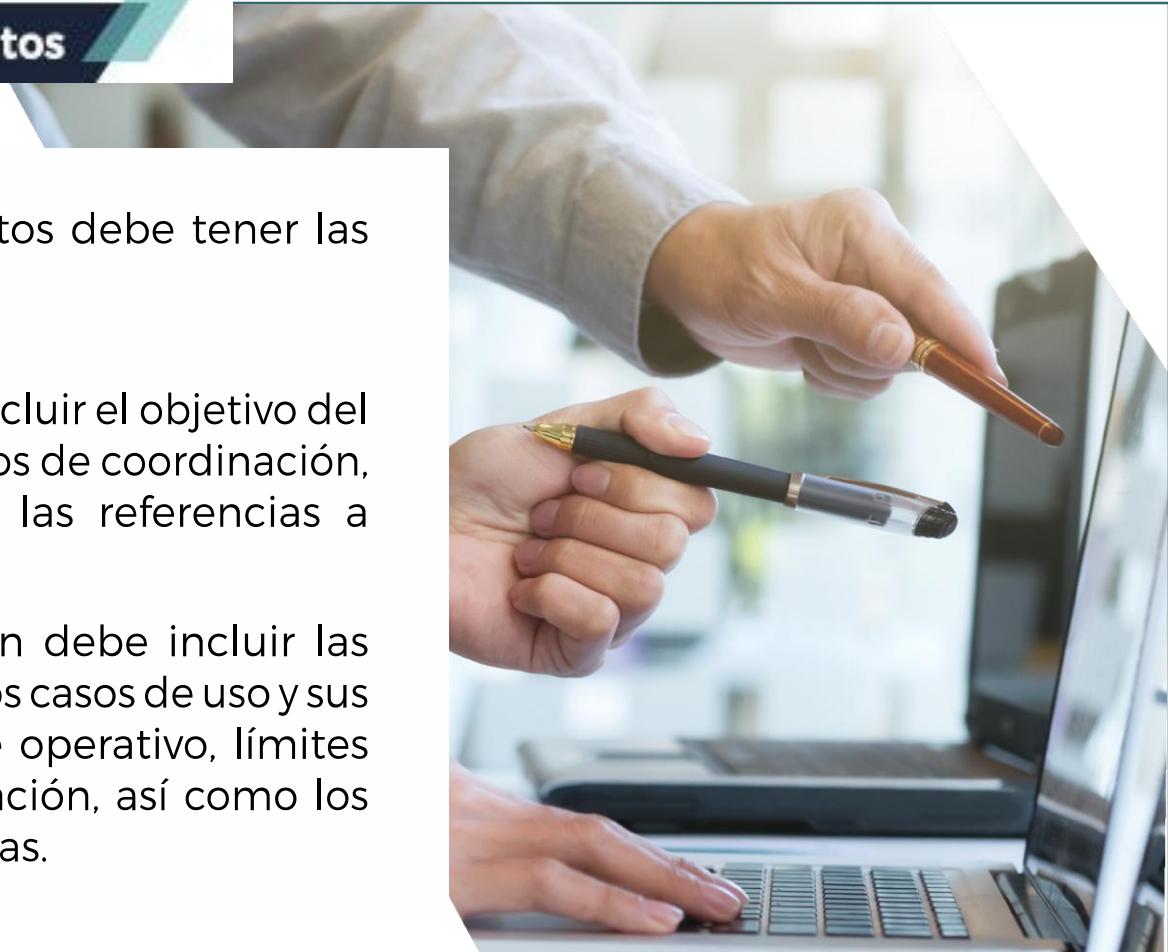
Un proyecto de software puede documentarse de distintas formas, entre las que se encuentran:

- Lenguaje natural escrito de manera consciente y bien estructurado.
- Modelos visuales que ilustren los procesos de transformación, los estados de sistemas y los cambios entre esos estados, así como las relaciones entre los datos, los flujos lógicos y similares.
- Especificaciones formales que utilicen las matemáticas para clarificar los requerimientos.



El documento de requerimientos debe tener las siguientes **secciones**:

- **Introducción.** Esta debe incluir el objetivo del sistema, los acuerdos internos de coordinación, el alcance del proyecto y las referencias a considerar.
- **Descripción.** Esta sección debe incluir las perspectivas del producto, los casos de uso y sus características, el ambiente operativo, límites del diseño y la implementación, así como los supuestos y las dependencias.





- **Características del sistema.** Aquí se especifican cada una de las características del sistema y sus requerimientos funcionales.
- **Requerimientos de datos.** Esta sección debe especificar el modelo de datos lógico, el diccionario de datos, los reportes que se generarán, los procesos de adquisición, integridad, retención y disposición de datos.

- **Requerimientos de interfaces.** Aquí se especifican las interfaces de usuario, las de software, las de hardware y las de comunicación.
- **Atributos de calidad.** En esta sección se especifican la usabilidad, el desempeño, la seguridad y cualquier otro estándar o mejor práctica que se requiera.
- **Apéndice A.** Glosario.
- **Apéndice B.** Modelos de análisis.





Lo anterior define la organización de un documento de requerimientos. Aunado a esto, cada elemento mencionado debe incluir **requerimientos pertinentes** con las siguientes características:

- **Completo.** Cada requerimiento tiene que contener toda la información necesaria para que cada lector pueda entenderlo. Si algo no se tiene hasta el momento, se debe escribir **TBD** (*to be determined*), para identificar esta ausencia y completarla después.

- **Correcto.** Cada requerimiento tiene que ser preciso y describir una capacidad que satisfaga alguna necesidad del cliente. De igual manera, debe dejar en claro la funcionalidad por construir.
- **Factible.** Cada requerimiento debe ser posible dentro de los límites técnicos, de presupuesto y de tiempo.
- **Necesario.** Cada requerimiento debe provenir de alguna fuente de autoridad. Por tanto, puede ser rastreado hasta quien lo solicitó.



Documentación de Requerimientos



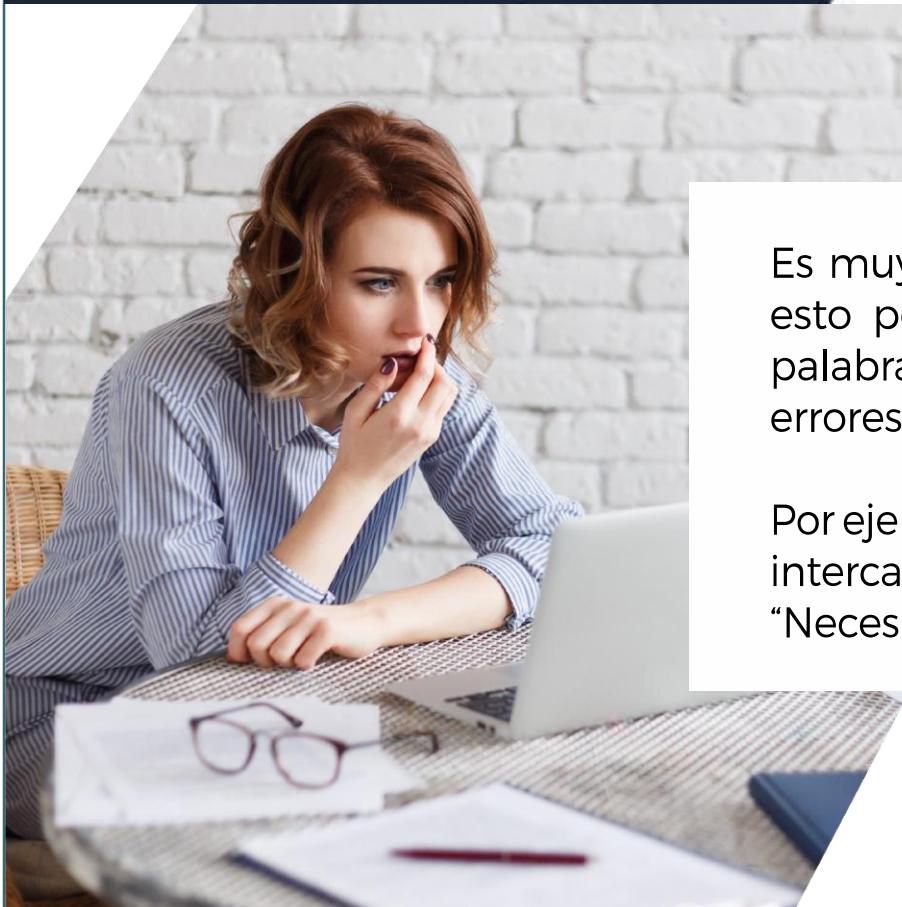
- **Priorizado.** Cada requerimiento debe tener un nivel de prioridad bien definido y argumentado. Este indicador debe ser establecido de manera colectiva involucrando distintas perspectivas.
- **Inequívoco.** Cada requerimiento debe evitar ser interpretado de dos o más maneras.
- **Verificable.** Cada requerimiento debe indicar cómo se comprueba que ha sido completado y es exitoso.

Documentación de Requerimientos

A grandes rasgos, **dos objetivos primordiales** deben cumplirse para decir que un requerimiento es bueno:

- Que varias personas lean el requerimiento y lleguen a la misma interpretación.
- Que la interpretación a la que lleguen sea aquella deseada por el autor del requerimiento.





Es muy importante **siempre ser consistente**, ya que esto permite que el lector esté seguro de que las palabras realmente significan lo que se indica y no son errores o frases *al aire*.

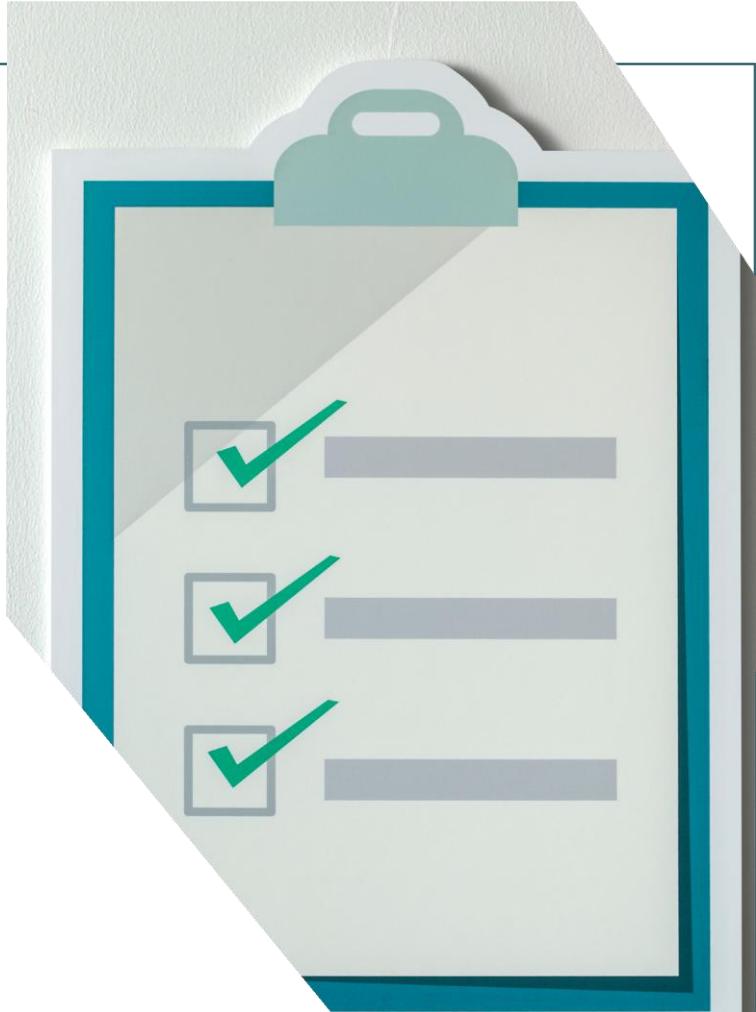
Por ejemplo, si en los requerimientos utilizas de manera intercambiada las siguientes palabras: “Debe”, “Tiene”, “Necesita” y/o “Puede”, el lector se confundirá.

Por tanto, un requerimiento siempre tiene que incluir la palabra correcta de manera consistente. En la sección de requerimientos funcionales, por ejemplo, podríamos redactar la siguiente frase:



“El sistema tiene que permitir la entrada de dos variables”.

De esta forma queda claro que **no es una opción**. Es algo que el sistema **tiene que hacer**.





Por el contrario, en la sección de **requerimientos deseables**, por ejemplo, redactaríamos la siguiente frase:



“El sistema debe permitir la entrada de dos variables”.

El verbo *deber* implica que algo puede o no ocurrir. De esta forma queda claro que **es una opción**. Es algo que el sistema **puede o no incluir**.

Por otra parte, **debes escribir los requerimientos de manera activa:**



El sistema tiene que sumar las variables A y B”.

En lugar de manera pasiva:



Las variables A y B se tienen que sumar por el sistema”.

Finalmente, deberás escribir **requerimientos individuales**. De otra manera, habrá problemas si una parte del requerimiento se cumple y la otra no.





Foro 1

Entorno de trabajo

Participa en el foro enviando imágenes que demuestren que ya tienes acceso a las siguientes herramientas en su versión de prueba:

- SpiraPlan

Presiona el botón para participar en el foro.



Conclusión

Durante esta unidad se identificaron técnicas de elicitación de requerimientos y las mejores prácticas para su documentación. Para ello se señalaron las secciones del documento de requerimientos y los componentes que debe contener.

En definitiva, es muy importante rescatar la idea de que los requerimientos son más que frases o un contrato. El documento de requerimientos tiene valor al promover que se hagan preguntas que de otra manera no se hubieran realizado y al generar un entendimiento común entre los involucrados.





¡Felicidades!

Acabas de concluir la **primera unidad** de tu curso *Redacción Avanzada*. Te invitamos a finalizar este esfuerzo realizando el examen parcial correspondiente. Para ello, debes regresar a la pantalla principal y dar clic en *Presentar examen*.



Unidad 2

Tipos de Requerimientos a Redactar



Temario Unidad 2

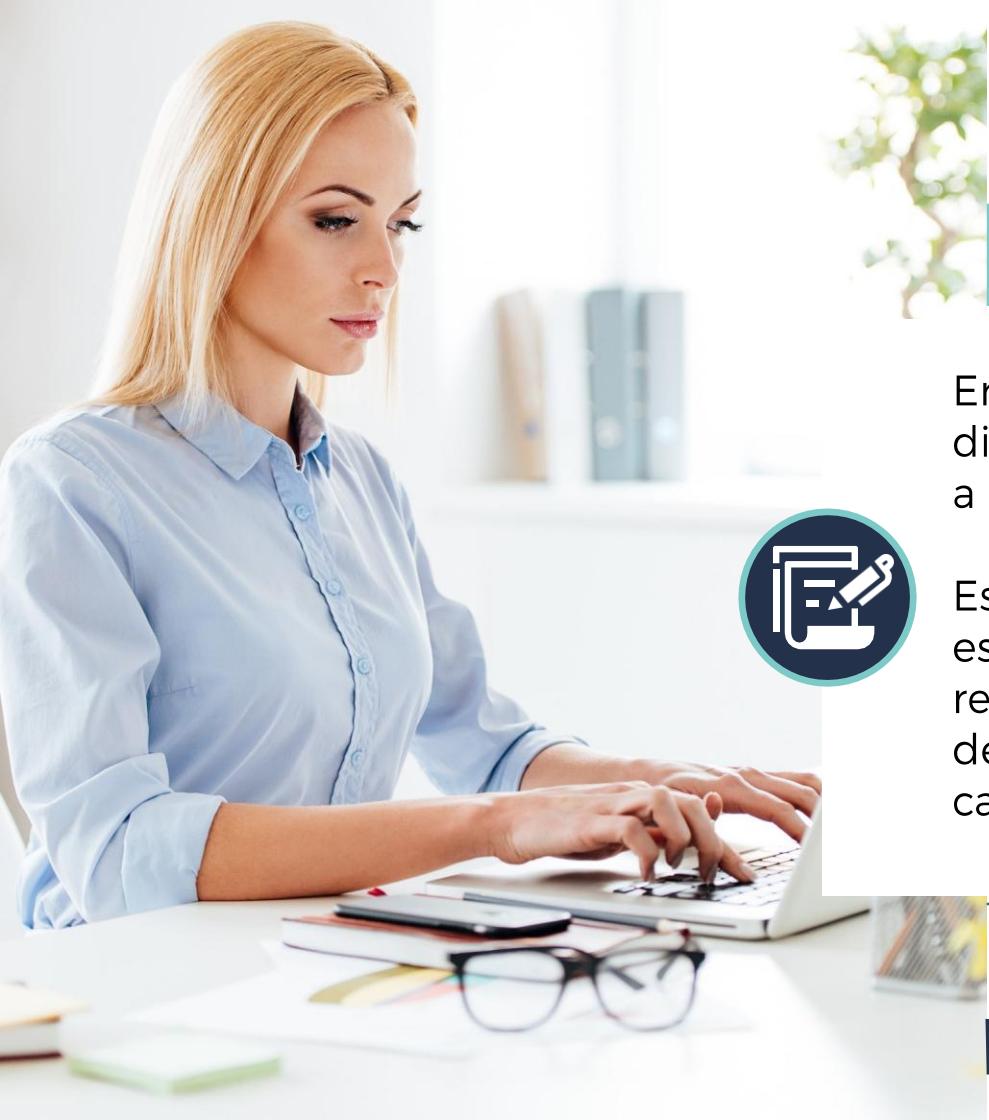
2.1

Requerimientos
Funcionales



1.2

Documentación
de Requerimientos



Introducción

En esta segunda unidad aprenderás a distinguir entre dos tipos de requerimientos, a saber: los funcionales y los no funcionales.



Esta distinción te permitirá no solo la estructuración de tu documento de requerimientos, sino también la identificación de las características que tiene que tener cada tipo.

Competencias a desarrollar



El alumno será capaz de definir y evaluar los requerimientos funcionales.



El alumno será capaz de definir y evaluar los requerimientos no funcionales.





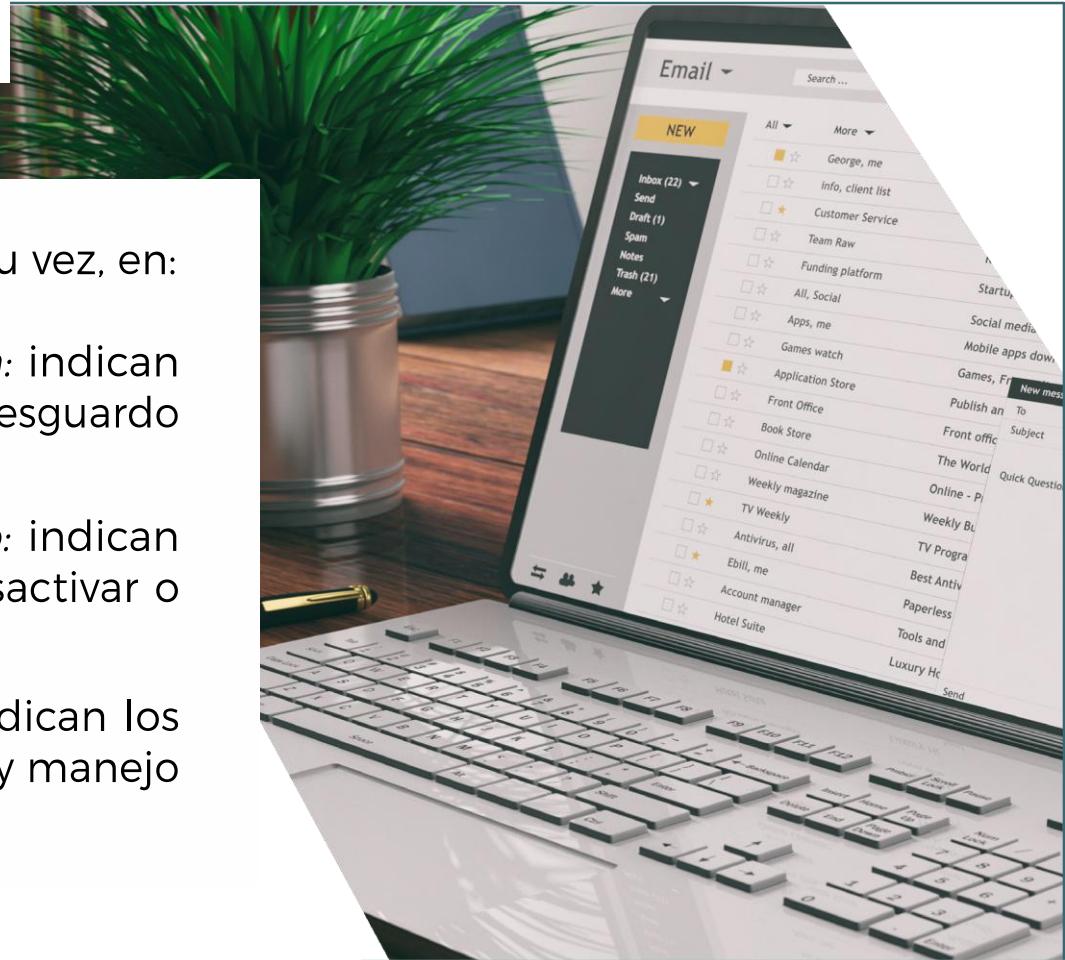
Requerimientos Funcionales

Un requerimiento funcional se enfoca en lo que el sistema tiene que hacer. Es decir, cómo se tiene que comportar. Los principales tipos de requerimientos funcionales son los siguientes:

- **Reglas de negocio.** En estos requerimientos debes definir la información que debe ser creada, leída, actualizada, borrada, buscada, enlazada, enviada, recibida o reportada.

- **Transacciones.** Se dividen, a su vez, en:

- *Transacciones de entrada:* indican los criterios para permitir el resguardo de información.
- *Transacciones de cambio:* indican los criterios para borrar, desactivar o cancelar una transacción.
- *Errores de transacción:* indican los criterios para la verificación y manejo de errores.



Requerimientos Funcionales



● **Funciones administrativas.** Describen aquellas funciones que los administradores de sistemas deben seguir para mantener en funcionamiento el sistema. Por ejemplo: desactivar o activar cuentas, preferencias del sistema, respaldos y monitoreo del sistema.

● **Autenticación.** Aquí se establecen los mecanismos que se utilizarán para verificar la identidad de los usuarios. Un ejemplo de esto es la necesidad de datos biométricos.

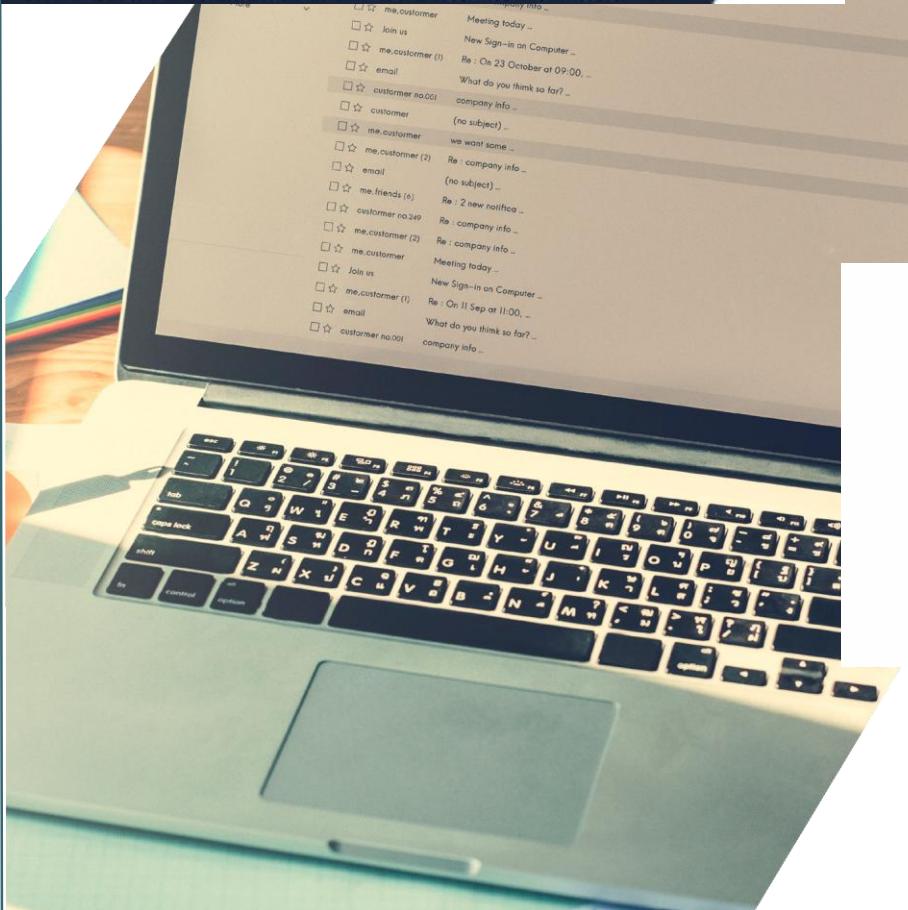
- **Auditoría.** Estos requerimientos se enfocan en dejar rastro de todas las acciones que conducen usuarios específicos. En particular, aquellos que tengan más poder de acción dentro del sistema.
- **Interfaces externas.** Aquí se especifican los sistemas externos de donde se obtendrá o a donde se enviará la información de nuestro sistema.



● **Certificaciones.** Son aquellos requerimientos que provienen de fuentes externas, como organismos internacionales. Se trata de estándares o mejores prácticas. Estas son importantes porque son requerimientos que el cliente probablemente nunca pida porque los desconoce. Sin embargo, cuando solicita “un software seguro”, esta seguridad se debe implementar a través de un estándar.



Requerimientos Funcionales



- **Búsquedas y reportes.** Estos requerimientos indican los operadores *booleanos* a utilizar; es decir, si los usuarios podrán ingresar texto directamente o solo podrán elegir opciones predefinidas. De igual manera, se establecen los diseños e indicadores que se producirán.

- **Cumplimiento regulatorio.** Similar a la sección de certificaciones, en este apartado se ingresan todos los requerimientos que están asociados a una entidad externa. En este caso, la entidad puede ser la legislación nacional o internacional. Estos no son opcionales y, por tanto, el resto de los requerimientos debe estar alineado a esta sección.



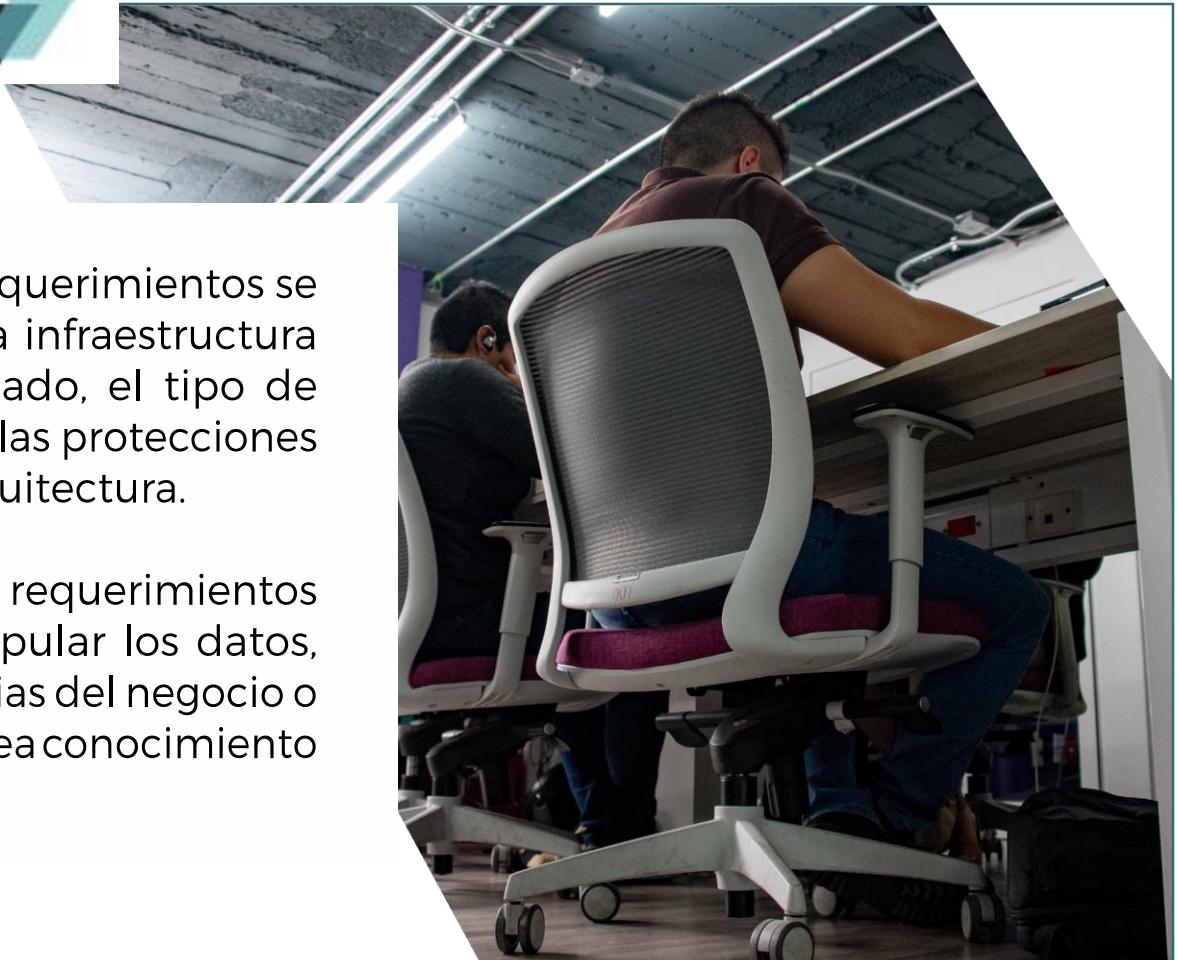
Requerimientos Funcionales



● **Datos históricos.** Aquí se especifican todos los requerimientos que permitirán el almacenado, procesamiento y aseguramiento de las bases de datos con las que la organización ya cuenta.

● **Archivo.** En esta sección se coloca en qué momento se pasará una información a la sección de archivo. Es decir, estará fuera de los principales dispositivos de almacenamiento.

- **Estructurales.** En estos requerimientos se establece cómo debe ser la infraestructura física. Eso incluye el cableado, el tipo de servidor, el centro de datos, las protecciones a desastres, así como la arquitectura.
- **Algoritmos.** Este tipo de requerimientos especifica formas de manipular los datos, fórmulas matemáticas propias del negocio o cualquier otro método que sea conocimiento propio del cliente.





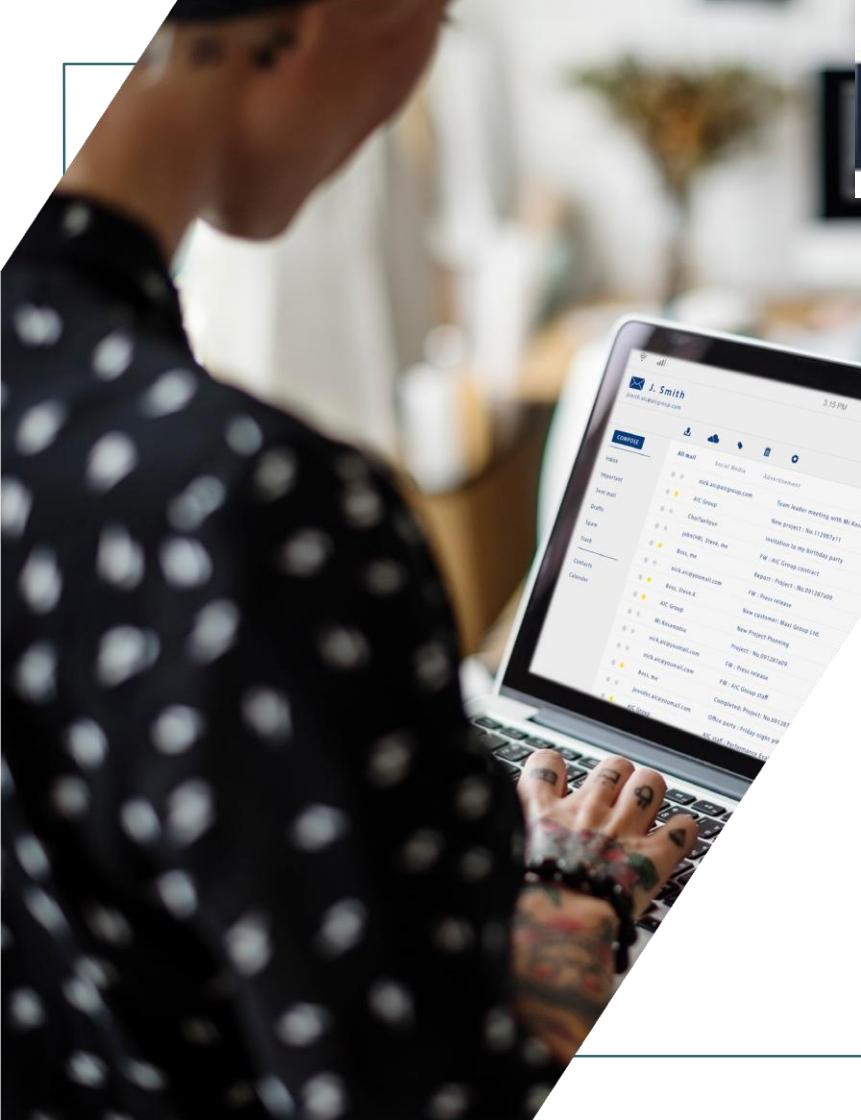
- **Base de datos.** Se establecen los tipos de datos y formatos de los principales campos utilizados por el cliente.
- **Respaldos y procedimiento de recuperación.** Estos requerimientos indican cómo se realizarán los respaldos y, en caso de que el sistema colapse, se pueda reestablecer.



Video

Te invitamos a ver el
siguiente video:





Requerimientos no Funcionales

Por su parte, **los requerimientos no funcionales no especifican lo que se tiene que hacer**. Algunos ejemplos son los siguientes:

- **Arquitectura.** Esta ya la tiene la compañía del cliente y, por tanto, se debe respetar. Independientemente del sistema del cual se esté hablando.

- **Capacidad.** Se establecen las capacidades de almacenamiento del sistema.
- **Restricciones.** Estos requerimientos pueden estar enfocados a la privacidad o cualquier otra preferencia de tipo ético o de presupuesto.
- **Documentación.** Se establece la documentación que será entregada como parte del sistema, por ejemplo: manuales de usuario y manuales técnicos.





- **Eficiencia.** Estos requerimientos se enfocan en la eficiencia de la energía.
- **Efectividad.** Estos requerimientos indican qué tan buenas deben ser algunas funciones del sistema. Por ejemplo, para algunas funciones es suficiente identificar una tendencia (60%, 70%, 80%, 90%). Sin embargo, en otras es importante calcular los datos de manera exacta y sin sesgos.

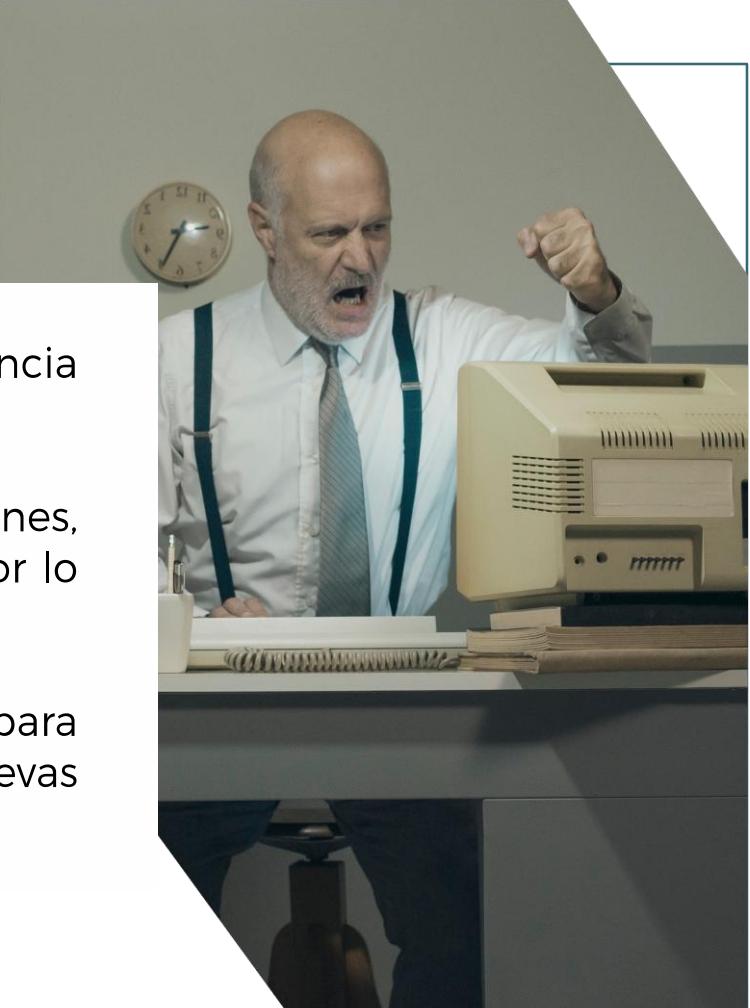
- **Tolerancia a fallos.** Aquí se especifica lo que acontece cuando una parte del sistema falla, pero no todo. Por tanto, se determina en qué ocasiones el sistema continuará funcionando, y en cuáles se parará.
- **Calidad.** Existen dos niveles que se deben cuidar:
 - Calidad a nivel de requerimiento.
 - Calidad a nivel de la integración de todos los requerimientos.





- **Integridad de los datos.** Son requerimientos enfocados en asegurar la integridad de los datos a lo largo de todo el ciclo de vida de los proyectos.
- **Tiempos de respuesta.** Se especifican los tiempos en que debe ejecutarse una transacción.
- **Reusabilidad.** Se busca que los requerimientos se establezcan de tal manera que puedan volver a utilizarse en nuevos proyectos.

- **Carga de trabajo.** Aquí se establece la concurrencia de procesos y cómo se reaccionará ante fallos.
- **Desempeño de acuerdo a perfiles.** En ocasiones, algunos perfiles necesitan mejor desempeño, por lo que esto debe especificarse.
- **Escalabilidad.** Indica la capacidad del sistema para seguir creciendo en el tiempo, de acuerdo a nuevas funcionalidades que aún no se conocen.



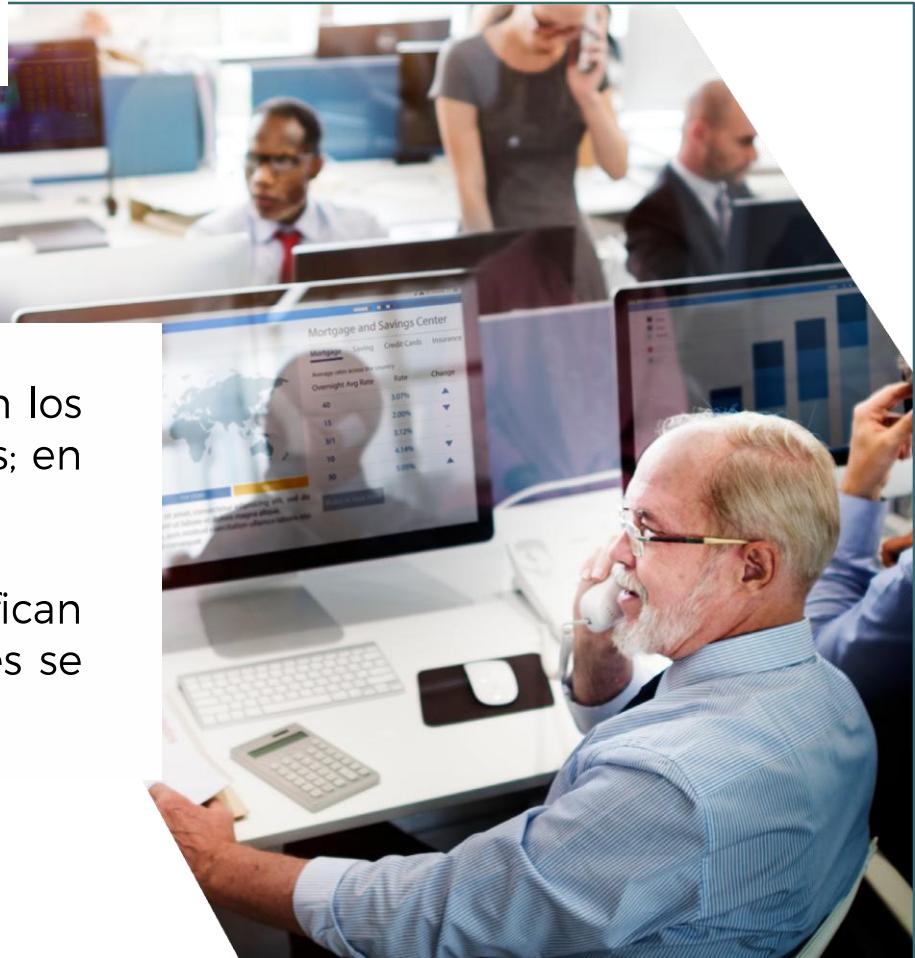
Requerimientos no Funcionales



- **Usabilidad.** Estos requerimientos se enfocan en solicitar elementos que hacen más sencilla la adaptación del usuario con el sistema.
- **Accesibilidad.** En esta sección se ingresan los requerimientos que permiten que el sistema pueda ser utilizado por la mayor cantidad de personas posibles, incluyendo personas con capacidades diferentes.

Requerimientos no Funcionales

- **Interoperabilidad.** Aquí se establecen los requerimientos para intercambiar datos; en particular sus formatos.
- **Portabilidad.** En esta sección se especifican las diferentes plataformas en las cuales se podrá ejecutar el software.





Actividad 1

Te invitamos a realizar la siguiente actividad:

Visualiza el siguiente video de apoyo:

Presiona el botón para descargar la Actividad:

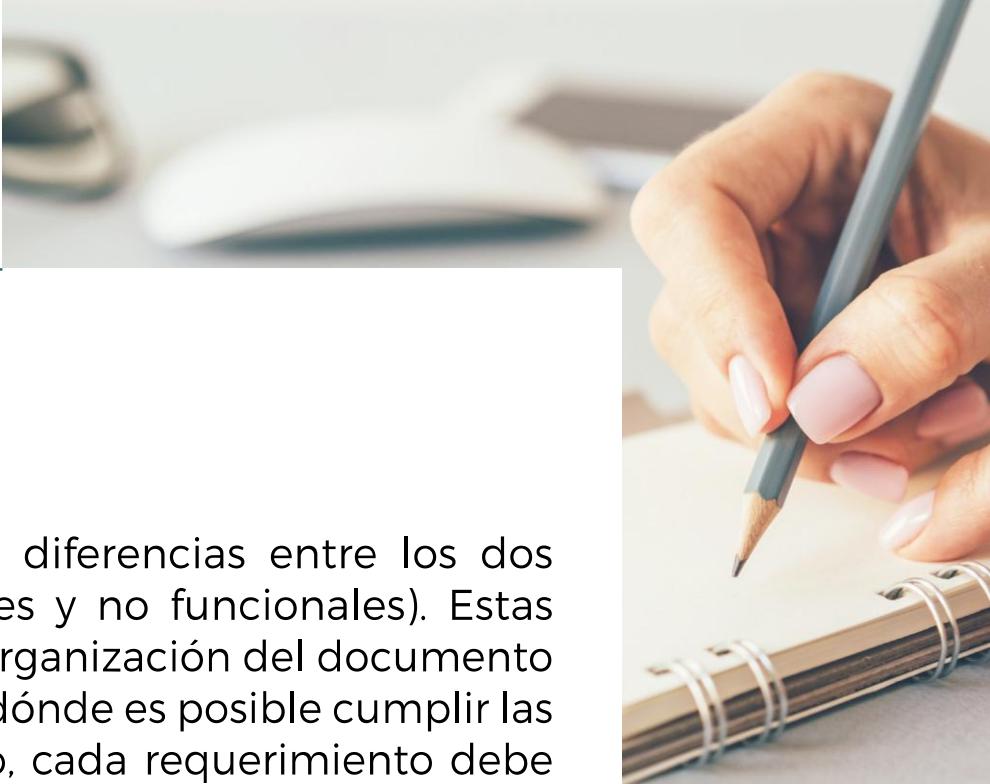
Presiona el botón para entregar la actividad:



Conclusión



Durante esta unidad se definieron diferencias entre los dos tipos de requerimientos (funcionales y no funcionales). Estas diferencias son importantes para la organización del documento final, pero también para saber hasta dónde es posible cumplir las expectativas del cliente. Por lo tanto, cada requerimiento debe estar asociado a uno de los dos tipos.





¡Felicidades!

Acabas de concluir la **segunda unidad** de tu curso *Redacción Avanzada*. Te invitamos a finalizar este esfuerzo realizando el examen parcial correspondiente. Para ello, debes regresar a la pantalla principal y dar clic en *Presentar examen*.



Unidad 3

Deficiencias en la Redacción de Requerimientos



Temario Unidad 3

3.1

Evitando
Ambigüedades



3.2

Evitando
Insuficiencias





Introducción

En esta tercera unidad aprenderás a identificar cuándo un requerimiento es ambiguo o está incompleto. Esta distinción te permitirá comunicar mejor al resto de la organización las necesidades del software o, en caso de que seas tú quien reciba el requerimiento, poder hacer preguntas clave desde un inicio y no cuando el software esté por terminarse.

Competencias a desarrollar

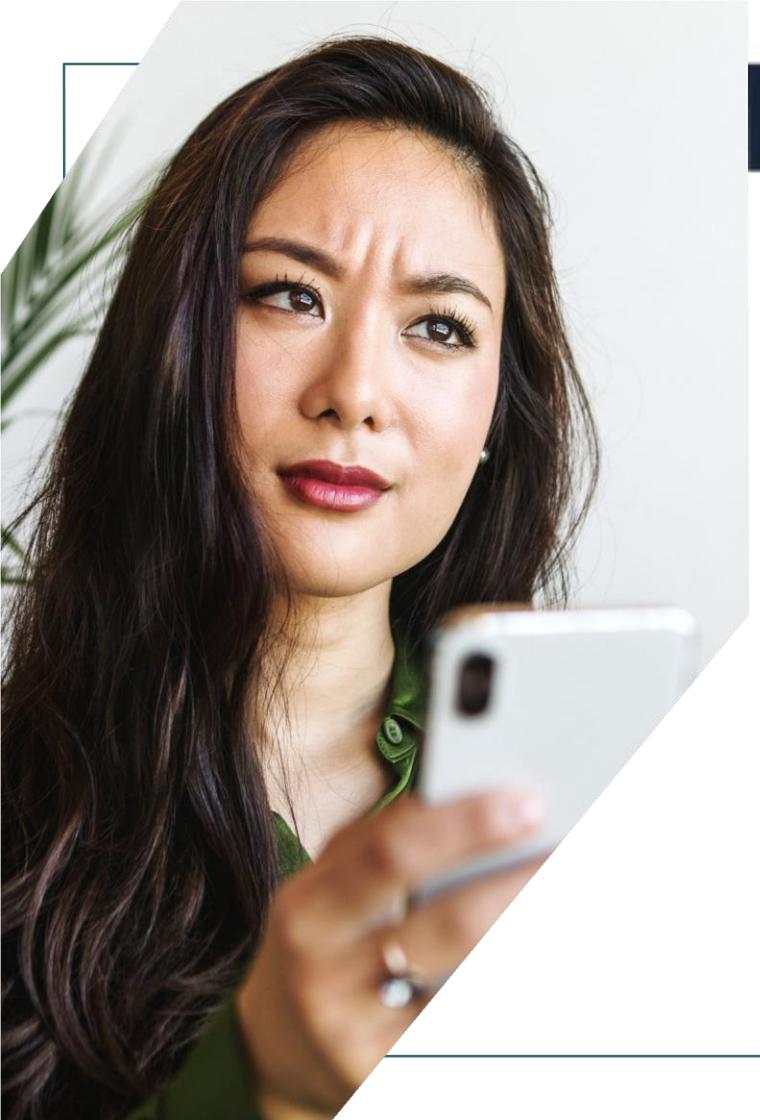


El alumno será capaz de definir cuándo un requerimiento de software es ambiguo.



El alumno será capaz de definir cuándo un requerimiento de software no está completo.





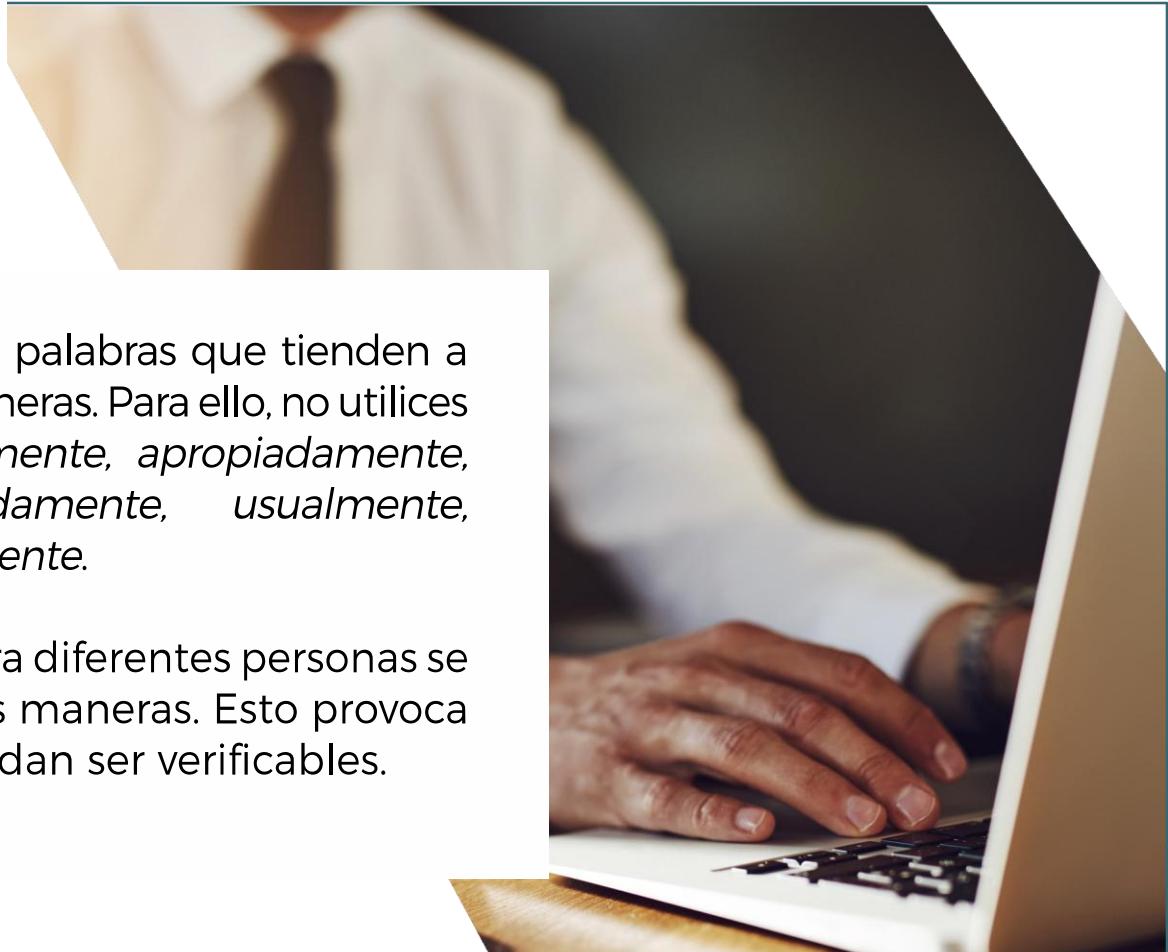
Evitando Ambigüedades

Quien recibe un requerimiento da cuenta de su calidad. En este sentido, si quien lo redacta considera que está todo claro, pero los lectores consideran que no es así, entonces el requerimiento no está bien redactado. Uno de los motivos para que esta situación suceda es la **ambigüedad de los requerimientos**, la cual puede surgir en las siguientes situaciones:

- **Palabras confusas.** Los términos deben utilizarse tal y como son definidos en el glosario. Nunca utilices diferentes palabras para referirte a lo mismo.

- **Palabras subjetivas.** Evita palabras que tienden a interpretarse de distintas maneras. Para ello, no utilices palabras como: *razonablemente, apropiadamente, generalmente, aproximadamente, usualmente, sistemáticamente y rápidamente*.

La principal razón es porque para diferentes personas se pueden interpretar de distintas maneras. Esto provoca que los requerimientos no puedan ser verificables.



Evitando Ambigüedades



Además, otras palabras que se tienen que evitar son: *aceptable, adecuado, mejor, eficiente, rápido y flexible*.

Cada una de las palabras anteriores es susceptible de causar confusión. Por ello, es mejor definir claramente los valores esperados.

Por ejemplo, en lugar de decir que algo es rápido, es mejor decir que *su velocidad debe ser de 1 000 MB por segundo*.

Evitando Ambigüedades

A parte de definir los resultados con valores esperados, es muy importante considerar palabras como “Y” y “O”. Ya que su significado puede impactar en el proyecto.

De igual manera, no agregue frases como: *por ejemplo, incluyendo, tales como*, entre otras.

En consecuencia, se deben de enlistar todos los valores de una función, no solamente ejemplos. O en su caso, referir al lector a la lista completa de valores.

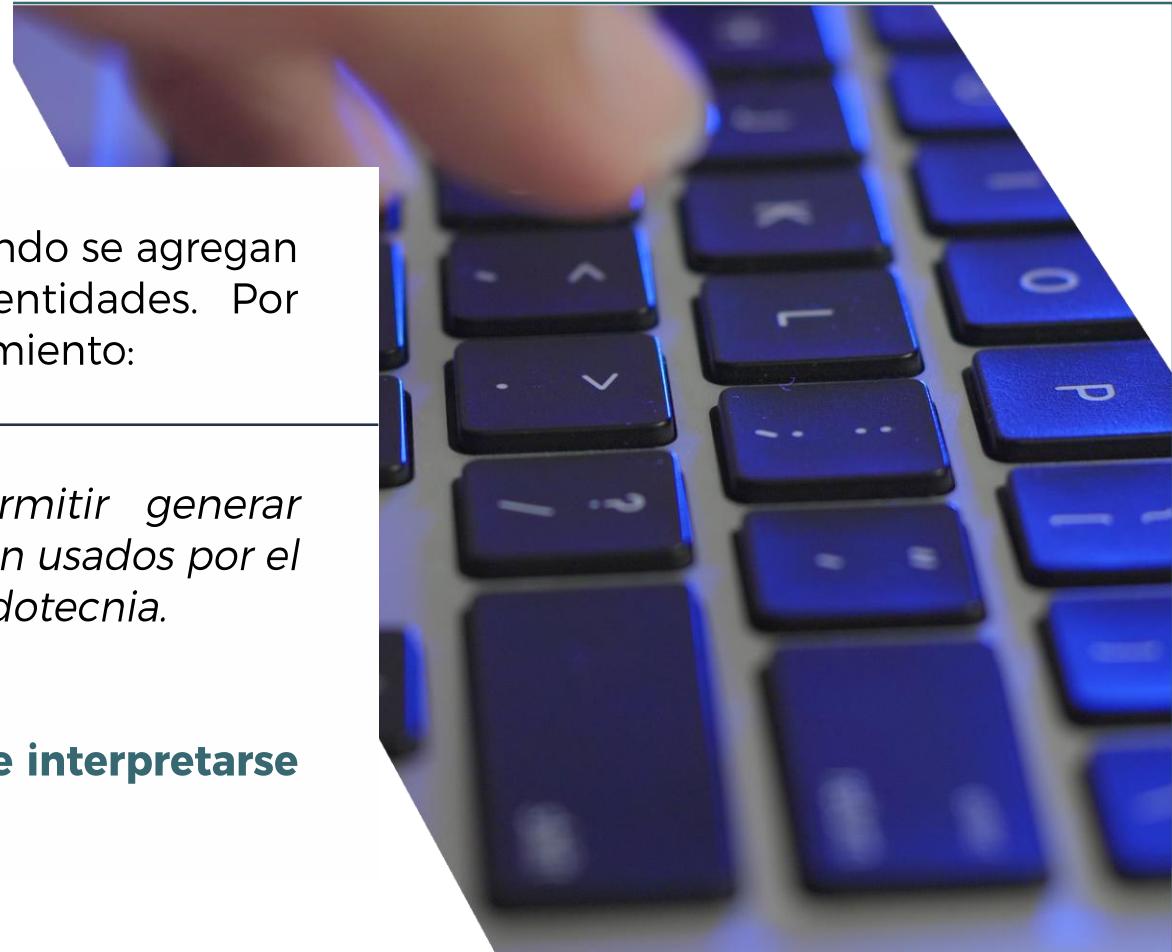


Evitando Ambigüedades

Un caso en particular se da cuando se agregan diagonales “/” para separar entidades. Por ejemplo en el siguiente requerimiento:

El sistema tiene que permitir generar correos masivos para que sean usados por el departamento Ventas/Mercadotecnia.

El requerimiento anterior **puede interpretarse de las siguientes maneras:**





- El nombre del departamento es *Ventas/Mercadotecnia*.
- *Ventas* y *Mercadotecnia* son sinónimos.
- Algunos le llaman al departamento *Ventas* y otros le llaman *Mercadotecnia*.
- Ya sea *Ventas* o *Mercadotecnia* pueden enviar correos masivos. Es decir, la diagonal significa “O”.
- Tanto *Ventas* como *Mercadotecnia* pueden enviar correos masivos. Es decir, la diagonal significa “Y”.

Evitando Ambigüedades

En ocasiones, se llega a este tipo de requerimientos porque el mismo redactor tiene dudas. Por consiguiente, esto provocará que los lectores también las tengan, y se generen problemas futuros durante el desarrollo o las pruebas.

Recuerda, cuando no estés seguro de algo, es importante que coloques la frase TBD, que en inglés significa: “*To Be Defined*”, y en español: “Será definido”.





Video

Te invitamos a ver el
siguiente video:





Evitando Insuficiencias

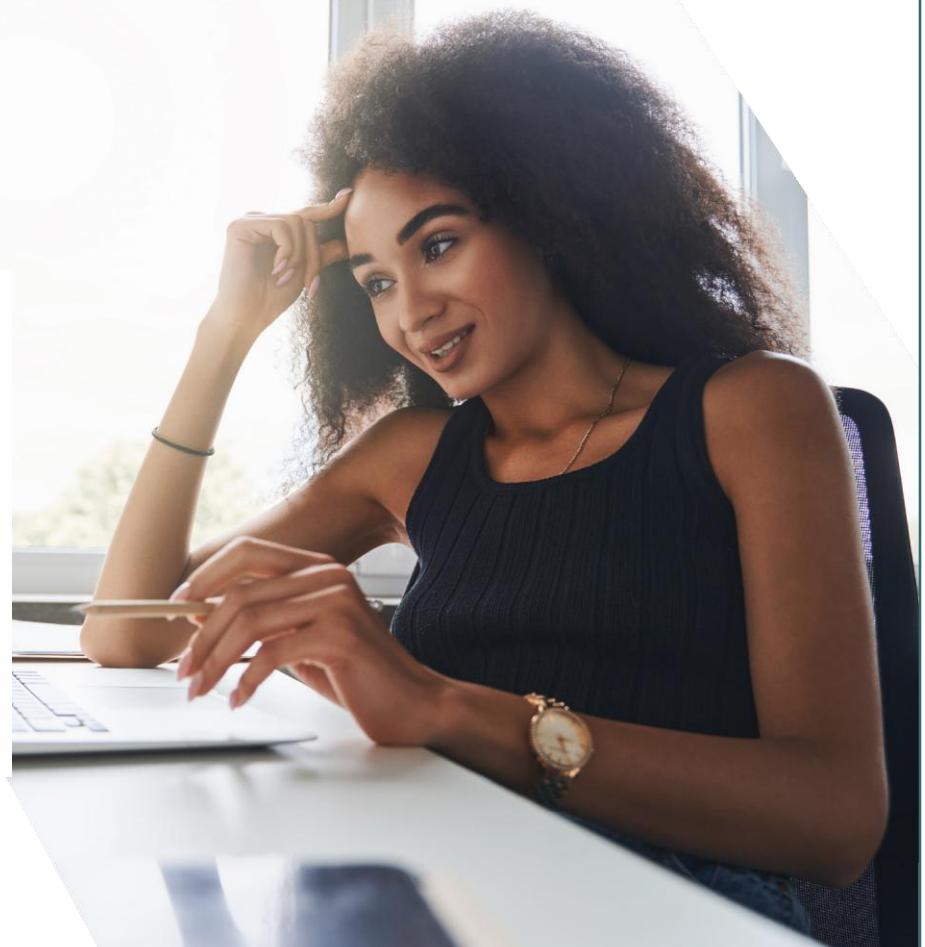
No es realmente posible saber si todo lo que se requiere se ha documentado en algún requerimiento. Sin embargo, una buena práctica es enfocarse en las tareas de los usuarios en lugar de una lista de requerimientos.

Las siguientes situaciones ayudan a entender lo que significa que **un requerimiento esté incompleto**:

- **Asimetría.** Esta situación se da cuando se solicita un requerimiento, pero no se especifican todos los que son necesarios para poder cumplir el primero. En este caso no se sabe si los requerimientos complementarios están perdidos o se asumieron.



● **Lógica compleja.** Los requerimientos se tienen que escribir de manera directa. De esta manera, en un solo requerimiento no se deben mezclar dos ideas o más. Por tanto, el redactor tiene que asegurarse que todas las indicaciones sean explícitas y que no haya nada implícito.



Evitando Insuficiencias



- **Excepciones perdidas.** Todos los requerimientos tienen que establecer cómo actuará el sistema en caso de que algo no esperado suceda. No es necesario saber exactamente lo que pasará. Sin embargo, es necesario que se especifique un plan para cuando algo no ocurra como se espera.

Cuando un requerimiento es fallido por cualquiera de los motivos antes expuestos, **lo mejor es reescribirlo.**

Idealmente, este debe ser partido en requerimientos más cortos y simples, así como frases inequívocas que tengan una base de evidencia comprobable.

Esto permite que se le pueda dar seguimiento a cada nuevo requerimiento.



Evitando Insuficiencias

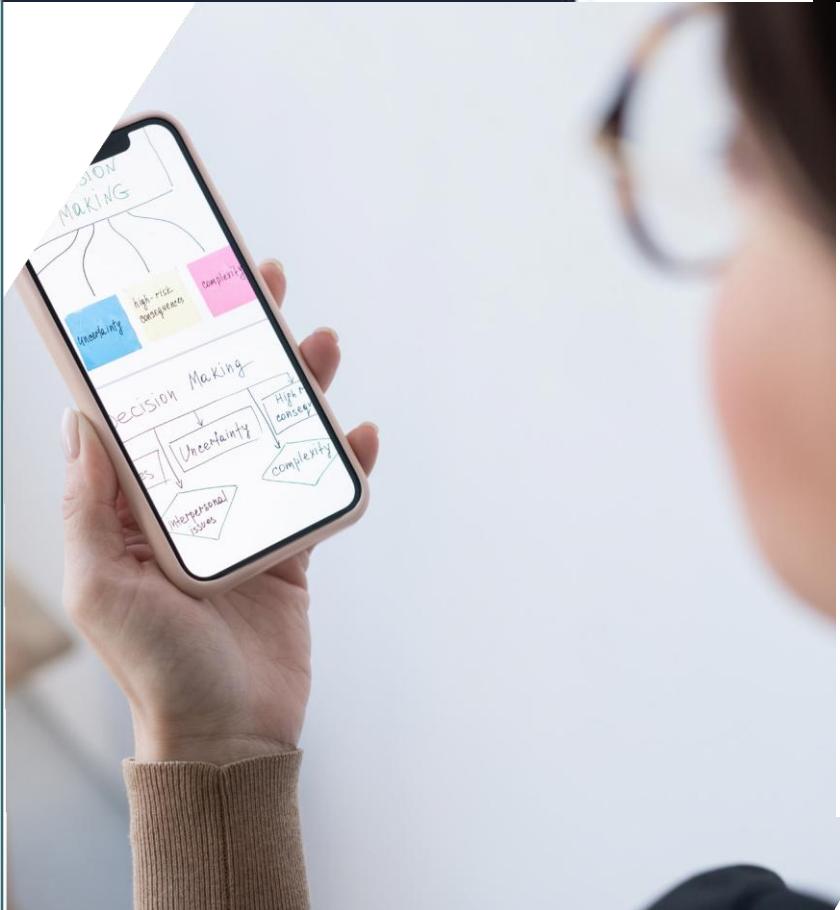


De igual manera, si el requerimiento es muy grande porque existe una lista de mensajes, es mucho mejor hacer uno al que se le agregue una tabla con los mensajes por desplegar de acuerdo a condiciones específicas. De esta manera, estará más claro renglón por renglón en qué caso es cuando se despliega cada mensaje.

Es importante señalar que **los requerimientos no deben escribirse de tal manera que limiten el diseño**. Así como hay insuficiencias de información, también puede haber excesos.

Los requerimientos tienen que estar enfocados en solicitar que se resuelva un problema. Sin embargo, deben permitir que se puedan decidir múltiples formas de resolverlos.



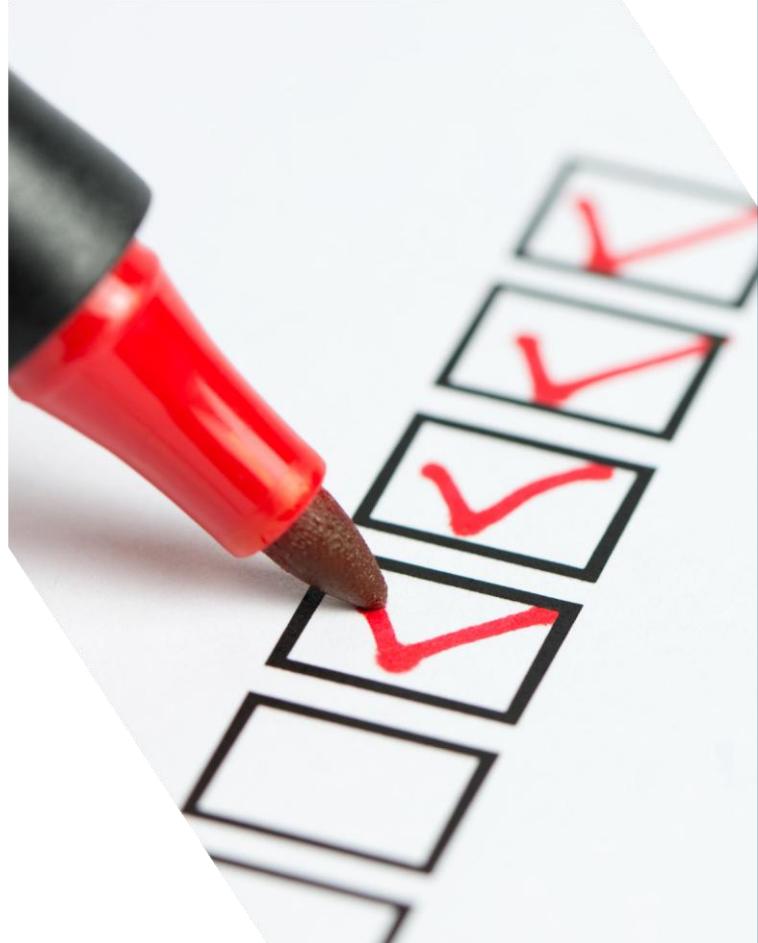


Por otro lado, las insuficiencias no solamente vendrán de la redacción. También pueden venir de los **elementos visuales**. Por tanto, siempre revisa los siguientes componentes:

- Diagramas de flujo.
- Diagramas de proceso.
- Diagramas de estado.
- Mapas de diálogo.
- Tablas o árboles de decisión.
- Tablas de evento-respuesta.
- Diagramas de caso de uso.
- Diagramas de actividades.
- Diagramas entidad-relación.

Un caso particular son los **reportes**. En ellos se tiene que especificar la siguiente información:

- Nombre.
- Propósito de negocio.
- Decisiones que se tomarán a partir del reporte.
- Forma de creación (automática / manual).
- Tamaño.
- Gráficas / Indicadores.
- Seguridad.
- Fuentes de datos.
- Parámetros y cálculos.
- Prioridad.
- Criterios de ordenamiento.





Además de la información anterior, es importante considerar los siguientes escenarios en cada reporte:

- **Variaciones.** Implican una diferencia en la secuencia de los datos, agregaciones o despliegue de información detallada. Cada una de estas variaciones debe indicarse claramente, ya que esto consume tiempo de desarrollo y, por tanto, requiere recursos.

Evitando Insuficiencias

- **Crecimiento.** No porque un reporte funcione con una pequeña cantidad de datos significa que lo hará con una cantidad de datos mayor. Por ejemplo, si se tienen dos departamentos, tres regiones o cuatro países, tal vez el reporte luzca bien. Pero al abrir una nueva sucursal o área, este podría no ser suficiente.

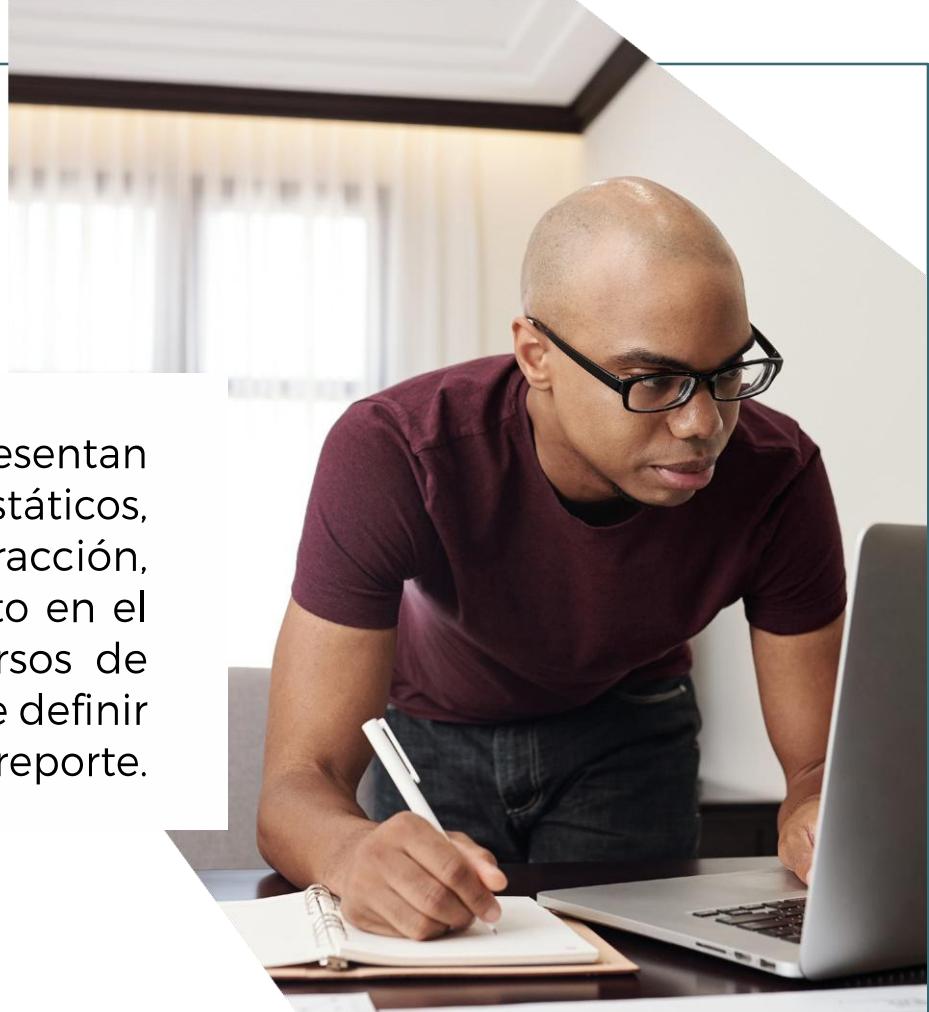


Evitando Insuficiencias



- **Similitudes.** Diferentes usuarios pueden requerir reportes similares. Sin embargo, no necesariamente se utilizan de la misma manera. Más allá de los temas de seguridad, la misma información puede representarse de distintas formas. Por ejemplo, horizontal o vertical. Además, es importante evaluar si se requieren distintos reportes o funcionalidades de visualización.

- **Dinamismo.** Los reportes dinámicos presentan datos en tiempo real; mientras que los estáticos, aunque tengas herramientas de interacción, presentan información de un solo punto en el tiempo. El dinamismo consume recursos de procesamiento, por lo que es importante definir de manera clara esta propiedad en cada reporte.



Evitando Insuficiencias



Otro caso particular es la instalación. En este sentido, **un software no está completo hasta que está instalado**. Por tanto, es importante definir lo siguiente:

- Proceso de la instalación inicial.
- Recuperación de una instalación incompleta, incorrecta o abortada.
- Reinstalación de la misma versión.
- Proceso para regresar a una versión anterior.
- Instalación de componentes adicionales o actualizados.
- Desinstalación.



Actividad 2

Te invitamos a realizar la siguiente actividad:

Visualiza el siguiente video de apoyo:

Presiona el botón para descargar la Actividad:

Presiona el botón para entregar la actividad:



Conclusión

Como pudimos estudiar a lo largo de la unidad, un requerimiento puede parecer bien redactado ante los ojos de quien lo escribe. Sin embargo, cada lector tiene intereses y necesidades distintas. Además, el contexto e información que cada lector posee puede diversificar el significado de las palabras o crear preguntas que no se habían pensado. Rara vez es posible poner a todos los involucrados a leer la lista de requerimientos, pero seguir de manera formal las prácticas descritas en esta unidad ayudará a disminuir los conflictos.





¡Felicidades!

Acabas de concluir la [tercera unidad](#) de tu curso [Redacción Avanzada](#). Te invitamos a finalizar este esfuerzo realizando el examen parcial correspondiente. Para ello, debes regresar a la pantalla principal y dar clic en [Presentar examen](#).



Unidad 4

Gestión de Requerimientos



Temario Unidad 4

4.1

Elementos
a Gestionar



4.2

Herramientas
para la Gestión



Introducción

En esta cuarta unidad aprenderás a coordinar la evolución de los requerimientos a lo largo de sus distintas etapas. Independientemente de la metodología de desarrollo de software que utilices, es importante que las solicitudes de los clientes se documenten de una u otra forma.

Esta documentación, junto a unos buenos procesos de gestión, permitirá que se dé un correcto seguimiento a cada requerimiento.



El alumno será capaz de identificar los elementos a gestionar en una lista de requerimientos.



El alumno será capaz de gestionar requerimientos mediante la herramienta apropiada.



Elementos a Gestionar

Una vez concluido el documento de requerimientos, es **importante que las personas, desde sus respectivos roles, lo revisen** para asegurar que se puede continuar. Estos roles son:

- **Autor.** Es la persona que creó y da mantenimiento al documento.
- **Moderador.** Es quien distribuye el material a evaluar y coordina las contribuciones del resto de los roles.

- **Lector.** Es la persona que parafrasea el requerimiento en las reuniones de revisión. Uno a la vez.

- **Documentador.** Utiliza formatos estandarizados para documentar los defectos encontrados durante la reunión.

Además de los roles anteriores, se invita a las reuniones de revisión a tantas personas involucradas en el proyecto como sea posible.





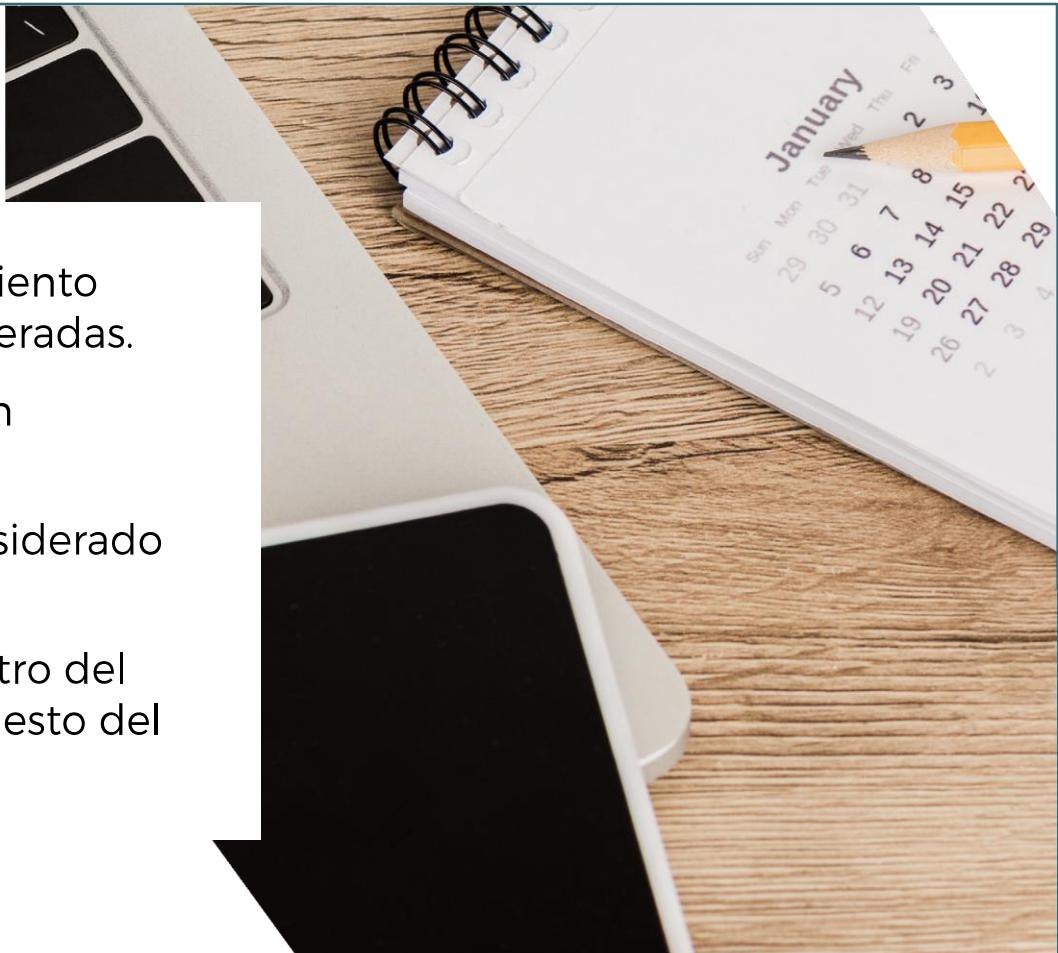
En una reunión de revisión se debe verificar que se cumplan las siguientes cuestiones:

- **Requerimientos Completos:**

- ✓ Se cumplen todas las necesidades conocidas del negocio.
- ✓ La información que se requiere se tiene a la mano. O, en caso contrario, se ha etiquetado como TBD.
- ✓ La comunicación entre equipos de hardware o software externos está definida.

Elementos a Gestionar

- ✓ Se ha definido el comportamiento esperado ante condiciones inesperadas.
- ✓ Los requerimientos se pueden priorizar, diseñar y probar.
- ✓ Cada requerimiento está considerado en una iteración y liberación.
- ✓ Cada requerimiento está dentro del alcance, calendario y del presupuesto del proyecto.



● Requerimientos Correctos:

- ✓ Los requerimientos no se duplican o contraponen.
- ✓ Los requerimientos son claros, concisos, sin ambigüedad y gramaticalmente correctos.
- ✓ Cada requerimiento puede ser evaluado.
- ✓ Cada requerimiento especifica los mensajes de error de manera clara y significativa.
- ✓ Los requerimientos son necesidades técnicas o de negocio. No son soluciones o restricciones.
- ✓ Los requerimientos son factibles.





● Requerimientos con Calidad:

- ✓ Los requerimientos especifican la usabilidad, desempeño y seguridad.
- ✓ Los requerimientos especifican las funciones que, al depender de la velocidad con la que se ejecuten, son críticas.
- ✓ Los requerimientos incluyen claramente cuándo una función se ejecutará de manera distinta, acorde a condiciones globales o locales.

● Requerimientos Organizados:

- ✓ Los requerimientos se documentaron de manera lógica.
- ✓ Las referencias entre requerimientos y hacia otros documentos son claras.
- ✓ Todos los requerimientos tienen un nombre apropiado y único.
- ✓ Cada requerimiento funcional puede ser rastreado hacia su regla de negocio.



Elementos a Gestionar



Otros:

- ✓ Los elementos visuales especifican todas las posibles excepciones o flujos alternos.
- ✓ Los requerimientos concuerdan con todos los elementos visuales.



Video

Te invitamos a ver el
siguiente video:



A photograph of a young woman with voluminous, curly reddish-brown hair. She is smiling warmly at the camera. She is wearing a dark blue, short-sleeved top. Her hands are resting on the keyboard of a silver laptop computer, which is open on a light-colored wooden desk. The background is a plain, light-colored wall.

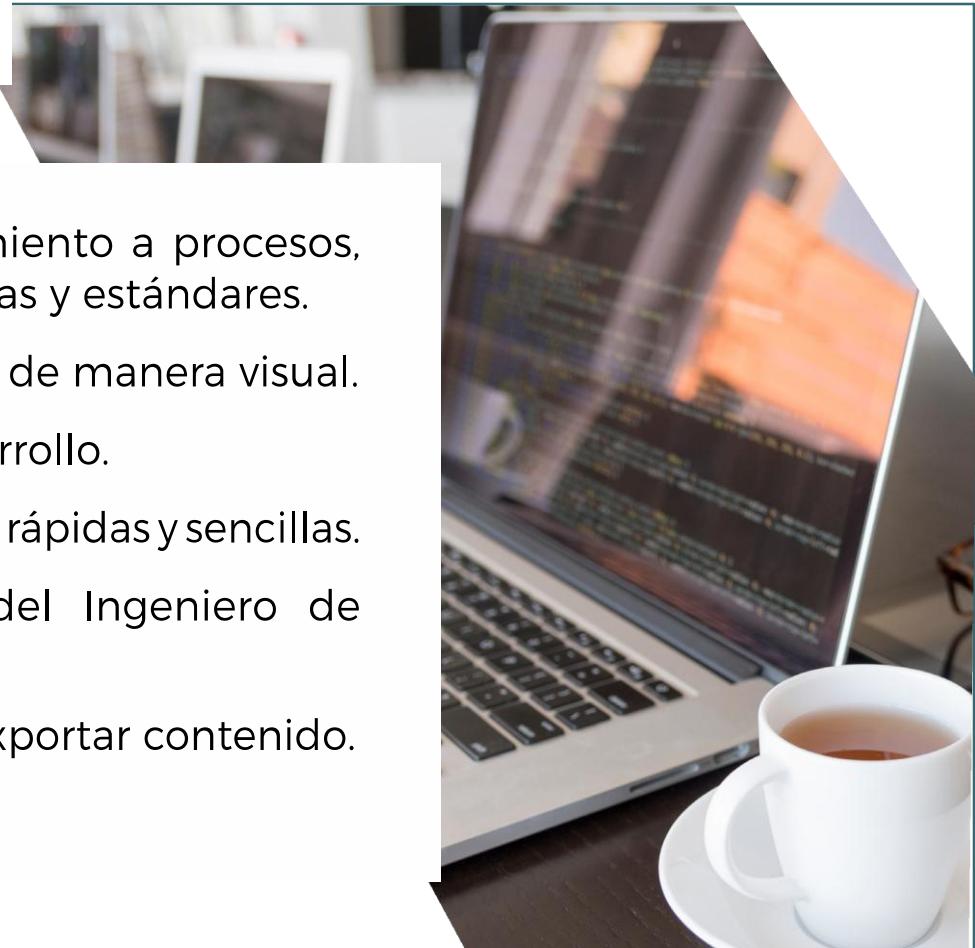
Herramientas para la Gestión

Tanto para documentar como para revisar los requerimientos, es necesario **llevar un orden**. Por ejemplo:

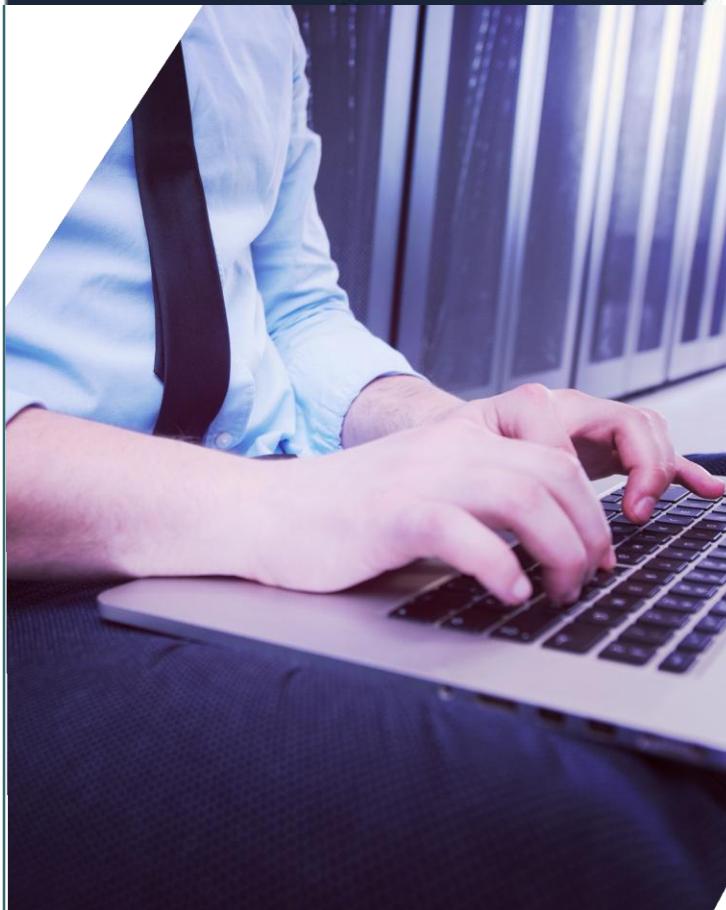
- A. Implementar una manera de dar seguimiento a los requerimientos y a lo que pasa con ellos.
- B. Entender cómo han evolucionado dichos requerimientos en el tiempo.
- C. Asociar los requerimientos a casos de estudio, historias de usuario o liberaciones.
- D. Dar seguimiento a las órdenes de cambios.

Herramientas para la Gestión

- E. Realizar una matriz para dar seguimiento a procesos, requerimientos, pruebas, dependencias y estándares.
- F. Representar la información requerida de manera visual.
- G. Soportar varias metodologías de desarrollo.
- H. Permitir una capacitación e instalación rápidas y sencillas.
- I. Evitar poner barreras a la labor del Ingeniero de requerimientos.
- J. Facilitar los medios para importar y exportar contenido.
- K. Dar un soporte de calidad.



Herramientas para la Gestión



Por otra parte, algunas de las **herramientas** que se recomienda revisar son:

- Jama Connect
- HelixRM
- Modern Requirements
- Orcanos
- Tample
- Visure
- Xebrio

Para ello, se debe realizar una matriz y comparar las características de cada herramienta para, de esta manera, elegir correctamente.

Software /Requerimiento

| | A | B | C | D | E | F | G | H |
|---------------------|---|---|---|---|---|---|---|---|
| Jama Connect | | | | | | | | |
| HelixRM | | | | | | | | |
| Modern Requirements | | | | | | | | |
| Orcanos | | | | | | | | |
| Tamplio | | | | | | | | |
| Visure | | | | | | | | |
| Xebrio | | | | | | | | |

Una vez que se haya elegido la herramienta, la siguiente **información** debe quedar clara por cada requerimiento:

- Identificador único.
- Fecha de creación.
- Historia de usuario o requerimiento asociado.
- Titulo.
- Versión.
- Autor.
- Prioridad.



Herramientas para la Gestión

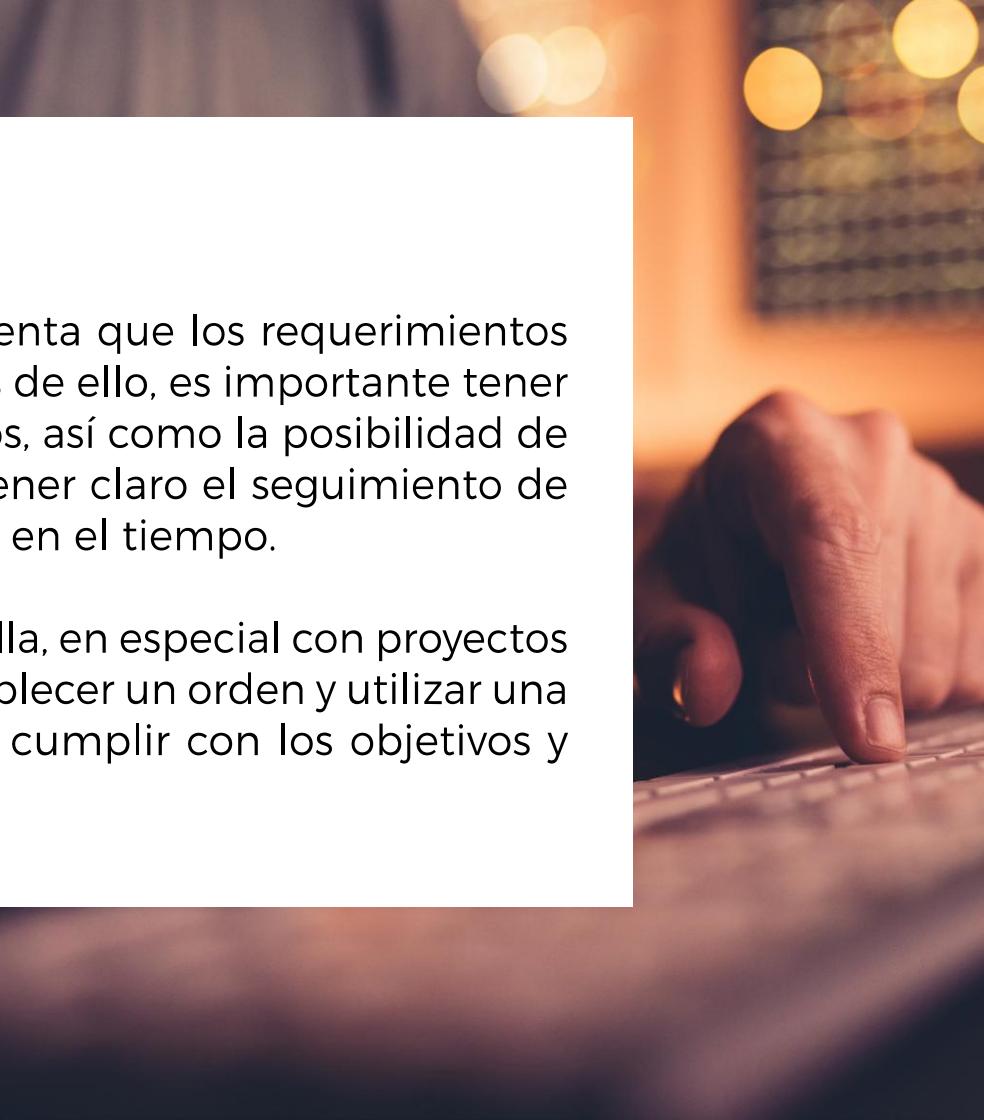


- Estado.
- Fuente.
- Riesgo.
- Asignación.
- Razones de inclusión/exclusión.
- Comentarios.
- Evidencia de éxito.
- Fecha límite.
- Fecha de última actualización.
- Requerimiento padre.
- Requerimientos hijos.
- Dependencias.
- Archivos adjuntos.

Conclusión

En esta unidad pudimos darnos cuenta que los requerimientos están en constante cambio. Además de ello, es importante tener claro que cada cambio genera costos, así como la posibilidad de nuevos conflictos. Por ello, es vital tener claro el seguimiento de esas solicitudes y cómo evolucionan en el tiempo.

En definitiva, la gestión no será sencilla, en especial con proyectos grandes. Por ello, es importante establecer un orden y utilizar una herramienta apropiada para poder cumplir con los objetivos y alcanzar las metas establecidas.





¡Felicitaciones!

Acabas de concluir la cuarta unidad de tu curso *Redacción Avanzada*. Te invitamos a finalizar este esfuerzo realizando el examen parcial correspondiente. Para ello, debes regresar a la pantalla principal y dar clic en *Presentar examen*.



Bibliografía



1. Koelsch, G. (2016). *Requirements Writing for System Engineering: Project success through realistic requirements.* Apress.
2. Wiegers, K. & Beatty, J. (2013). *Software Requirements.* Microsoft.



PROYECTO FINAL

A close-up photograph of a person's fingers typing on a dark blue computer keyboard. The keyboard is illuminated with a vibrant blue light, creating a glowing effect on the keys. The background is blurred, showing more of the keyboard and the person's hands. In the lower-left corner of the slide, there is a dark teal graphic element. A thin white horizontal line extends from the bottom of the word 'FINAL' in the text area to a small white dot on the teal graphic, creating a visual connection between the text and the image.

Unidad 4.

Gestión de requerimientos

Te invitamos a realizar el proyecto final:

Visualiza el siguiente video de apoyo:



Presiona el botón para descargar el proyecto final



Presiona el botón para entregar el proyecto final

