Los requerimientos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema (o un sistema de software) para que tenga valor y utilidad para el usuario. En otras palabras, los requerimientos muestran qué elementos y funciones son necesarias para un proyecto.

En el modelo clásico de desarrollo de sistemas o desarrollo software, la etapa de los requerimientos viene antecedida de la etapa de factibilidad del sistema/software y precedida por la etapa de diseño del sistema/software.

**Etapas de la fase de requerimientos**

\* Obtención de requerimientos: búsqueda y obtención de los requerimientos desde los grupos de interés.

\* Análisis: comprobación de la consistencia y completitud de los requerimientos.

\* Verificación: constatación de que los requerimientos especificados son correctos.

**Clasificación de los requerimientos**

\* Requerimientos funcionales: qué debe hacer el sistema o software.

\* Requerimientos no funcionales: cómo debe funcionar el sistema o software (no su implementación), por ej. calidad, rendimiento, facilidad de uso, etc.

\* Requerimientos externos: a qué se debe atener el sistema o software con respecto a su entorno: compatibilidad con otros sistemas, adecuación a determinadas leyes, etc.

**Características que deberían cumplir los requerimientos**

\* Actual: el requerimiento no debe volverse obsoleto con el paso del tiempo.

\* Cohesión: el requerimiento debe dirigirse a solo una única cosa.

\* Completo: el requerimiento debe estar completamente declarado en un único lugar, sin información faltante.

\* Consistente: el requerimiento no debe contradecir ningún otro requerimiento y debe ser completamente consistente con toda la documentación.

\* Correcto/necesario: el requerimiento debe cumplir con la necesidad declarada por los interesados en el sistema/software.

\* Factible/viable: el requerimiento debe poder ser implementado.

\* No ambiguo: el requerimiento debe estar concisamente declarado. Debe expresar hechos objetivos, no opiniones subjetivas. Debe poder poder ser interpretado de una única manera.

\* Obligatorio: el requerimiento debe representar una característica definida por el grupo interesado en el desarrollo del sistema/software, su ausencia no puede ser reemplazada.

\* Observable externamente: el requerimiento debe especificar una característica observable externa o experimentable por el usuario del producto.

\* Verificable/demostrable: La implementación del requerimiento debe poder ser resuelta en alguno de estos cuatro métodos: inspección, análisis, demostración o prueba.

Niveles de granularidad

### **Nivel de usuario**

El requerimiento a nivel de objetivo de usuario es aquel que:

* Abarca una única tarea bajo la responsabilidad de un único individuo;
* Es realizado en el momento en que el usuario posee todo lo que es necesario para que la tarea sea completada hasta el límite de su responsabilidad en el flujo operativo donde está contenida.

Al final de la tarea el usuario alcanza su propósito, está satisfecho y no hay nada más que necesite hacer. Una vez que el requerimiento fue completado, todo lo que debería de realizarse para tratar un evento externo fue realizado. Esta tarea es casi siempre parte de un proceso de negocio que puede tener un flujo operativo más amplio y por esto puede aún no estar completa. Sin embargo, la perspectiva relevante en este caso no es del proceso de negocio, es de la tarea.

En general, si un trabajo involucra más de un individuo es porque hay más de una tarea presente en el contexto. Hay excepciones como un retiro en la cuenta corriente en la sucursal del banco donde dos individuos participan: la caja del banco que solicita e informa datos para el retiro y el cliente que informa la contraseña.

Esta es la granularidad del ejemplo 2 (“Transferir una cifra de una cuenta corriente para otra cuenta corriente.”). Es una única tarea (seguramente compuesta por varios pasos y reglas), bajo la responsabilidad de un único individuo que al final de todos los pasos está satisfecho con el objetivo alcanzado: una cifra fue transferida a otra cuenta corriente.

### **Nivel agregado**

El requerimiento en este nivel agrega varios requerimientos a nivel de objetivo de usuario en una única especificación de más alto nivel. Tanto más alto sea su nivel, más generales son sus objetivos y para que un objetivo de nivel más alto sea alcanzado, otros objetivos de nivel más bajo deben ser alcanzados primero.

Este tipo de requerimiento está relacionado a objetivos más generales y su amplitud está asociada a objetivos colaborativos o asociados a procesos de negocio de alto nivel. No es relativo a una única tarea, por lo contrario, agrega un conjunto de tareas de uno o más usuarios. Esta es la granularidad del ejemplo 1 (“Realizar transacciones con la cuenta corriente.”).

¿Cuáles son las tareas asociadas a este tipo de requerimiento? Tal vez sean obvias para los lectores de la especificación (los interesados) o tal vez aún no sean conocidas. Sin embargo, es sabido que hay actividades a realizar para levantar (refinar) este requerimiento. Lo que importa es que ya está claro que este requerimiento está dentro del alcance del software a ser desarrollado.

Sin embargo, algunos RF de este tipo poseen patrones que eliminan la necesidad de proporcionar más detalles. Un ejemplo clásico son los formularios CRUD (Create, Read, Update, Delete) para que el usuario pueda mantener datos por medio del software. Es muy común que esto sea expresado como: “El sistema debe mantener productos.”. Hay un conocimiento tácito entre los interesados que el verbo “mantener” agrega las tareas del CRUD. Por lo tanto queda claro a todos los lectores que el software debe permitir al usuario realizar las siguientes tareas: agregar, modificar, eliminar y consultar datos de producto.

### **Nivel de subfunción**

Estos requerimientos son pedazos de requerimientos con objetivo de usuario; están relacionados a un conjunto de pasos o a reglas de una o más tareas de los usuarios.

El requerimiento en nivel de subfunción que representa un conjunto de pasos describe el cambio de datos en los dos sentidos entre el usuario y el software; y entre el software y los requerimientos de almacenamiento. Este es el caso del ejemplo 3 (“Validar la tarjeta y la contraseña del cliente.”). Cada tipo de transacción que utiliza la cuenta corriente (ej.: retiro, transferencia, pago, etc.) exige el mismo conjunto de pasos descrito por el ejemplo 3, que se podría suponer como:

* Verificar si la tarjeta es válida.
* Verificar si la transacción elegida es compatible con el tipo de tarjeta.
* Verificar si la contraseña informada es correcta.
* Incrementar el control de errores de contraseña en caso que la contraseña informada sea incorrecta.
* Cambiar a cero el control de errores de contraseña en caso que la contraseña informada sea correcta.

Validar la tarjeta y contraseña del cliente no es el objetivo del cliente al utilizar el sistema de cajero automático, sin embargo son pasos necesarios y intermediarios para alcanzar su objetivo: por ejemplo, hacer un retiro.

El requerimiento en el nivel de subfunción que está relacionado a reglas, en general se vincula a las leyes que gobiernan el negocio y describen de manera que complementan los procesos de negocio. También llamadas muchas veces como reglas de negocio. La regla puede describir políticas corporativas, reglamentos gubernamentales o estándares de la industria por los cuales el software debe estar subordinado.

Este es el caso del ejemplo 4 (“Garantizar que la suma de todas las transacciones del cliente en el día no puede ser superior a $5,000.”). Las reglas de negocio muchas veces son compartidas entre varios RF, hasta entre distintos productos de software en la empresa.