

# Conceptos

## Sistema

conjunto de componentes que interactúan entre sí para alcanzar un objetivo global. Cada componente tiene un propósito y las interacciones están reguladas de alguna manera. Cada sistema tiene propiedades que emergen de las interacciones entre las componentes.

## Software

Es el conjunto de programas, datos y toda la documentación generada durante el desarrollo. El desarrollo de software es un proceso formado por etapas. Los nombres de las etapas y la manera en que se estructuran dependen del modelo elegido.

## Ciclo de vida

Empieza con la necesidad o la idea y termina cuando el sistema deja de utilizarse.

## Ámbito de desarrollo

- Inhouse: la org cuenta con un área de desarrollo en la cual el equipo construye los sistemas que serán utilizados por las otras áreas de la misma orga
- Empresa comercial: el equipo es un área de una organización que construye un sistema para el área de comercialización de la misma organización
- Contrato: una orga contrata los servicios de otra para que construya un sistema de acuerdo a las cláusulas del contrato
- Outsourcing: subcontrato - una orga contratada por otra para construir un sistema, contrata a una 3ra para desarrollar algún módulo de sw, o una empresa comercial que contrata a otra para construir 1 o más módulos

Características de las soluciones:

- Flexibilidad del sw para adaptarse a los cambios
- reutilizar componentes
- distribución instantánea

## Ingeniería de sw

El PROCESO de desarrollo de software se realiza en el marco de un PROYECTO, que se inicia para resolver un PROBLEMA. El proyecto aspira a obtener un PRODUCTO de calidad para un cliente. El resultado del proceso es un producto que contiene una o más componentes de software.

El PRODUCTO se construye mediante un PROCESO en el marco de un PROYECTO, para resolver un PROBLEMA. El problema puede entenderse como:

- Oportunidad de negocio
- Oportunidad de mercado
- Idea de producto
- Necesidad del negocio

La ingeniería de software es la disciplina que brinda y aplica los principios y las metodologías que conducen el proceso de desarrollo de software para fabricar un producto de calidad en el marco de un proyecto.

El desarrollo de software es una actividad creativa, disciplinada, colaborativa, guiada por principios y metodologías

Es un proceso sistemático con foco prioritario en la calidad.

## Metodología de desarrollo de software

Conjunto integrado de métodos, técnicas y herramientas que soportan el proceso de desarrollo de software.

### Flujos de proceso

Modelo en cascada - lineal - las etapas se realizan en secuencia

Iterativo - una o más tareas pueden repetirse antes de avanzar a la siguiente

Paralelo: no es un proceso, el orden en el que se realizan las tareas es irrelevante, y pueden hacerse en simultáneo

Evolutivo: el producto evoluciona a través de sucesivas versiones.

## Ingeniería de requerimientos

Área de la ing de sw que se ocupa del desarrollo y administración de requerimientos.

Algunos autores consideran la etapa del estudio de factibilidad como la inicial de la ingeniería de requerimientos, y otros la consideran fuera, siendo esta previa a la ingeniería.

Adoptamos el enfoque de que el estudio de factibilidad está por fuera de la ing de requerimientos.



El desarrollo y la administración de requerimientos, son actividades conducidas por el analista pero intervienen varios participantes: clientes, usuarios directos e indirectos, miembros del equipo, actores externos. Clientes y usuarios son fuentes de requerimientos.

Actores: cualquier entidad externa que interactúa con el sistema: usuarios directos (humanos) , sistemas de sw externos que interactúan con el producto (artificiales)

## Desarrollo de requerimientos

### Requerimiento

- Es algo que se pide o se solicita a alguien.
- Declaración que expresa una necesidad y las condiciones asociadas para satisfacerla.
- **Es una declaración que identifica una característica o restricción operativa, funcional o de diseño de un producto o proceso, que es inequívoca, comprobable, medible y necesaria para la aceptabilidad del producto o proceso.**

### Estructura del entorno

Tanto el administrador del proyecto como el ingeniero de requerimientos, deben conocer el entorno acerca del cual se desarrolla el producto. Entorno externo e interno a la organización, y el entorno operativo para el cual se desarrolla el sistema.

Ambos trabajarán en conjunto para definir los objetivos del negocio, el alcance del proyecto, los criterios de calidad para el producto, y los riesgos del proyecto.

Entorno externo: conceptos, regulaciones, oportunidades tecnológicas, condiciones de mercado, estándares de la industria



### La organización

El desarrollo de requerimientos implica aprender acerca del entorno de la aplicación: conocer su naturaleza y los beneficios que se esperan alcanzar.

- ¿Por qué existe?
- ¿Cuál es su propósito?
- ¿Qué servicios ofrece?
- ¿Qué obstáculos enfrenta?
- ¿Cómo se decide acerca del éxito o fracaso?

- ¿Cuál es su localización? ¿Dónde opera? ¿Por qué se elige cada localización?
- ¿Qué objetivos son generales y cuáles son específicos de cada localización?
- ¿Cómo es la estructura de la organización?
- ¿Cuáles son los valores fundamentales en relación al personal?

## El plan estratégico

No es responsabilidad del analista escribir el plan estratégico de la organización, pero cuando existe es indispensable conocerlo e indagar si efectivamente se está utilizando un marco para la toma de decisiones.

Habla de la naturaleza de la organización:

- Misión
- Visión
- Objetivos
- FODA (Fortalezas, Debilidades, Oportunidades, Amenazas)

La responsabilidad del analista está vinculada al producto en el marco de un proyecto que se desarrolla como parte de un proceso de negocio dentro de una organización que está afectada por el entorno externo.

La lectura e interpretación del plan estratégico permitirá:

- Comprender por qué se desarrolla el producto
- Durante el desarrollo de requerimientos, permitirá tomar decisiones sobre las características que el sistema debería exhibir o no exhibir
- Cuando el sistema está en producción, permitirá tomar decisiones acerca de los cambios y las propuestas de mejoras
- Al finalizar el proyecto, permitirá definir el nivel de éxito o fracaso del mismo, en el contexto de la misión y la visión de la organización, considerando los objetivos específicos del negocio.
- Establecer qué información debe brindar el sistema para medir el logro de los objetivos del negocio

## Proceso de desarrollo de requerimientos

Modelo iterativo incremental y metodologías ágiles. Son adecuados cuando el sistema se desarrolla en etapas y en la primera se obtiene una versión con algunas capacidades fundamentales.

El desarrollo de requerimientos es el inicio del proceso del desarrollo del sistema. Es también un proceso, que comienza cuando los clientes expresan sus primeras intenciones, acerca de un producto que en ese momento no existe, o no se ha identificado.

El analista CAPTURA las necesidades que dan origen al problema

Las ANALIZA

ESPECIFICA los requerimientos en documentos

VALIDA si los requerimientos especificados modelan efectivamente el problema a resolver



Este proceso se completa cuando se ha alcanzado el acuerdo base.

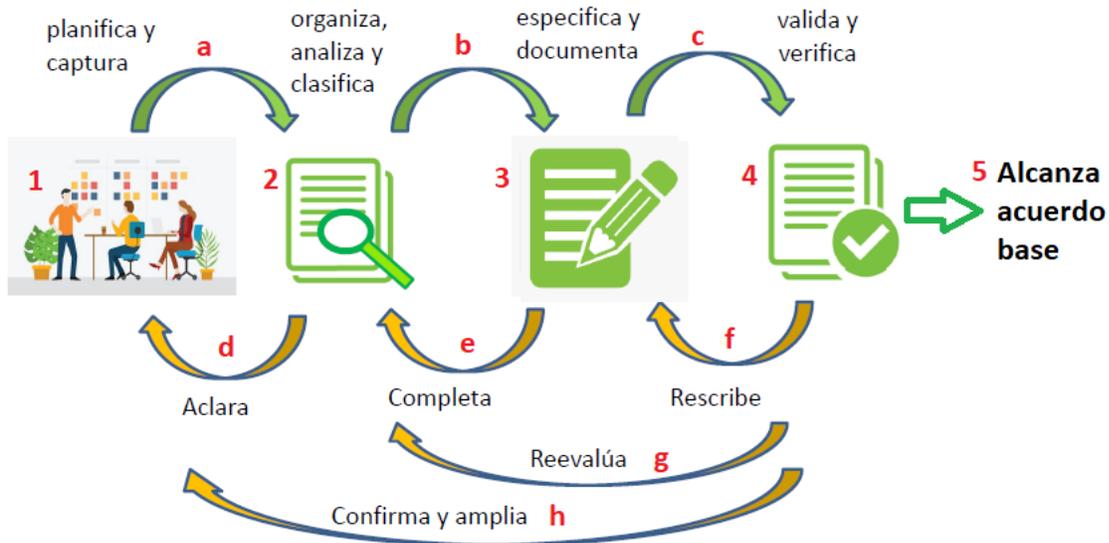
Los requerimientos brindan los fundamentos sobre los que se basan tanto el proceso de desarrollo del sistema, como la administración del proyecto.

## Modelos

Las actividades del desarrollo de requerimientos pueden organizarse de acuerdo a diferentes modelos:

### Refinamiento iterativo progresivo incremental

En cada etapa es posible avanzar o volver a alguna anterior.



1. Etapa de planificación y captura
  - a. Luego de la captura de los requerimientos, el analista organiza, analiza y clasifica los requerimientos capturados.
2. Etapa de organización, análisis y clasificación de los requerimientos capturados
  - b. Luego de finalizada esta etapa, el analista especifica y documenta los requerimientos
  - d. El analista puede volver a la etapa de captura para aclarar imprecisiones o resolver conflictos
3. Etapa de especificación y documentación
  - c. Luego de especificar y documentar los requerimientos, el analista pasa a la etapa de verificación y validación de los mismos.
  - e. En esta etapa, el analista puede retroceder a la etapa de análisis para completarlo
4. Etapa de verificación y validación

Al finalizar esta etapa, si todo salió como se esperaba, se alcanzará el acuerdo base y comenzará el diseño del sistema

  - f. El analista puede retroceder a la etapa de especificación y documentación para reescribir
  - g. El analista puede retroceder a la etapa de análisis para reevaluar
  - h. El analista puede retroceder a la etapa de captura para confirmar y ampliar la captura en una nueva sesión

## 5. Acuerdo base alcanzado

### Foco del desarrollo de requerimientos

#### Centrado en el usuario

El analista busca explorar y entender las tareas del usuario y de allí derivar el comportamiento del sistema que resultará valioso para esas tareas

Es importante que el analista tenga conocimiento acerca del entorno externo o interno de la organización (conocimiento del dominio, conocimiento del negocio, universo del conocimiento):

- Vocabulario,
- Objetivo, visión y misión de la organización,
- Estructura de la organización y su entorno
- Participantes o partes interesadas.

#### Centrado en el producto

El analista se concentra en las características que el sistema debería tener para atender una necesidad del negocio o de un segmento del mercado

### Captura de requerimientos

#### Participantes o partes interesadas (stakeholders)

El analista va a interactuar con todas las partes interesadas, tanto durante el desarrollo como durante la administración de los requerimientos



Es importante reconocer todas las clases de participantes, pero también no elegir demás de lo que sea necesario para capturar todos los requerimientos

- Líder de las decisiones: persona o grupo de personas que toman las decisiones o coordinan la toma de decisiones.
  - Criterios para la selección del líder:

- Ocupar un puesto relevante para la toma de decisiones
    - Tener conocimiento profundo del entorno o parte del entorno
    - Está expuesto a percibir conflictos o tiene intereses personales
    - Puede influir en la aceptación del producto o en el éxito del proyecto
  - Herramientas para la selección del líder
    - Organigrama
    - Catálogo de clientes
    - Segmento de mercado al que se aspira alcanzar
  - Las decisiones pueden ser:
    - Resolver un conflicto
    - Establecer prioridades
    - Aprobar requerimientos como acuerdo base o parte del acuerdo base
    - Aceptar cambios en los requerimientos acordados en un nivel o en el acuerdo base
  - Mecanismos de decisión
    - El grupo opina y decide el líder
    - El grupo discute hasta llegar a un acuerdo unánime y si no hay acuerdo decide el líder
    - El grupo vota y se decide por mayoría
- Clientes: son los que demandan el desarrollo del producto y por lo general esperan obtener un beneficio con el producto. Son fuente de requerimientos.
  - Patrocinadores y socios
  - Directivos, Gerentes, Ejecutivos
  - Auditores y asesores legales internos
  - Supervisores
- Usuarios: la participación de usuarios es indispensable cuando se adopta un enfoque centrado en el usuario
  - Directos: son quienes van a interactuar directamente con el producto. La implementación del producto puede o no resultarles beneficiosa de alguna manera. Son fuente de requerimientos.
  - Indirectos: reciben salidas del sistema, pero no interactúan directamente.
  - Clases de usuarios: es importante reconocer todas las clases de usuarios, ya que de no ser así pueden quedar requerimientos ausentes
    - Identificar clases de usuarios
      - Criterio para clasificar usuarios
        - Tareas que realizan / categoría de trabajo
          - Usuarios Operativos
            - contacto diario con el sistema
            - visión local
            - términos físicos
            - interesados en funcionalidad e interfaz
          - Usuarios Supervisores
            - están a cargo de los usuarios operativos
            - interesados en el aumento de productividad que puede dar el nuevo sistema
          - Usuarios Ejecutivos

- conocen las estrategias de mercado, expectativas de ganancias
    - visión global
    - familiarizados con modelos abstractos
    - interesados en la información que se pueda obtener del sistema
    - tienen influencia en las políticas de la empresa
  - Experiencia con relación a proyectos de software
    - Novato
    - Experto
  - Las características que usan del sistema
  - La frecuencia con la que usan el producto
  - Los niveles de seguridad o acceso a privilegios
  - La experiencia, expertitud y nivel jerárquico
  - Conocimiento en plataformas
  - Actitudes y características personales
  - Lenguaje nativo
- Seleccionar y trabajar con individuos representativos de su clase
  - Responsables de la selección de los usuarios
    - Patrocinador del proyecto
    - Dpto o área que establece el objetivo del negocio
    - Dptos o áreas a los que afectará el sistema
    - Dptos que interactúan con el contexto exterior
  - Mecanismos de selección
    - Brainstorming con tantas clases de usuarios como se puedan identificar
    - Identificar grupos con necesidades similares de manera tal que se puedan agrupar en una misma categoría más general
    - Usuarios que pertenezcan a 2 o más dptos
- Seleccionar un líder de usuarios por cada clase de usuarios
  - Cualidades del líder de los usuarios
    - Representa los intereses y necesidades de su clase
    - Conoce el entorno de aplicación
    - Conoce sus responsabilidades y funciones como líder
    - Tiene autoridad para tomar decisiones
    - Está motivado, comprometido y tiene autorización para dedicar el tiempo que demanda cumplir sus funciones
  - Funciones del líder de usuarios
    - Fuente de requerimientos
    - Colabora en la resolución de conflictos y definición de prioridades
    - Propone criterios para definir si el producto está terminado
    - Asume junto con el analista y el líder de decisiones, la responsabilidad de la calidad del producto

- Equipo de desarrollo
  - Administradores
    - Proyecto
    - Base de datos
  - Analistas: son quienes estarán en constante comunicación con clientes, a fin de dar parte de los avances. También es muy importante el compromiso de los clientes en el proceso de desarrollo, y que éstos tengan comunicación fluida con los analistas a fin de comunicarles cualquier cambio en el entorno interno o externo que pueda afectar a la satisfacción del cliente.
  - Arquitecto del sistema
  - Diseñadores
  - Desarrolladores
  - Testers
  - Instructores y capacitadores
  - Responsable de la atención de usuarios
- Externos
  - Asesores legales
  - Inversores
  - Proveedores
  - Contratistas
  - Auditores externos
  - Consultores y expertos externos
  - Agencias de gobierno
  - Sindicatos
  - Colegios profesionales

## Actores

Son usuarios humanos o no humanos que interactúan con el sistema.

Un actor representa un ROL para el sistema, una clase de usuario, mientras que el USUARIO es una persona específica.

Los usuarios artificiales, o no humanos, pueden ser:

- Agentes de software ejecutando un servicio en nombre del usuario.
- Sistemas con los que va a interactuar el producto

Algunos agentes pueden estar proscritos, en cuyo caso el sistema va a tener requerimientos para evitar que se satisfagan esas necesidades.

## Técnicas de captura basadas en encuentros (DR4 y DR7)

Especificar que tipo de requerimientos se van a capturar con cada técnica

Plan de captura para un encuentro (DR5)

Técnicas de captura basada en artefactos (DR6)

## Documentación de requerimientos

Una alternativa es generar un único Documento de Especificación de Requerimientos (SRS) que integre los requerimientos del negocio, los requerimientos del usuario, requerimientos funcionales, atributos de calidad, restricciones y requerimientos para los datos. El documento puede recibir otros nombres como Especificación del Sistema o Documento de Requerimientos del Producto.

Otra alternativa es generar dos documentos. El primero está orientado a los clientes y usuarios. Una vez validado este documento se genera otro para el equipo de desarrollo que probablemente irá evolucionando durante el diseño.

Una tercera posibilidad es generar el Documento de Visión y Alcance, el Modelo de Casos de Uso, el Modelo de Datos y el Modelo de requerimientos funcionales y no funcionales. Es decir, varios documentos organizados por niveles.

Cada alternativa tiene sus ventajas y desventajas. Cuando los requerimientos se distribuyen en varios documentos el mayor desafío es preservar la coherencia, sobre todo cuando hay modificaciones. También puede resultar difícil identificar en qué momento las especificaciones de casos de uso dejan de aludir a las tareas del usuario y comienzan a describir funcionalidades. Con frecuencia el modelo va evolucionando progresivamente y la combinación de especificaciones detalladas y representaciones visuales permite describir la funcionalidad.

En todos los casos las reglas del negocio se especifican en otro documento y se etiquetan de manera que pueden ser referenciadas desde el SRS.

## Documento de especificación de requerimientos (SRS)

El SRS establece las funciones y capacidades que el sistema de software debe brindar, sus características y las restricciones que debe respetar. Describe el comportamiento esperado del sistema ante diferentes condiciones y las principales cualidades que debe exhibir.

El SRS recibe otros nombres según la organización y también dependiendo de los niveles de requerimientos que especifica:

- Documento de Requerimientos
- Especificación Funcional
- Especificación del Producto
- Especificación del Sistema

Cualquiera sea el nombre, este documento integrado es el punto de partida para el diseño, codificación y verificación, aunque no se anticipa especificando aspectos de ninguna de estas etapas. La expectativa no debería ser escribir el SRS del producto completo antes de comenzar el desarrollo, pero sí capturar los requerimientos de cada iteración, antes de comenzar a diseñar las funcionalidades que corresponden a esa iteración. En un modelo de proceso secuencial el SRS está completo en el momento que comienza a trabajar el diseñador. En un modelo incremental se establece el acuerdo base y el SRS se va completando en paralelo a las etapas que siguen. El cliente no debería esperar que el sistema brinde una funcionalidad o exhiba una cualidad que no está especificada en el SRS.

### Destinatarios

Los destinatarios del SRS son:

- El administrador del proyecto que elabora el plan del proyecto
- Los desarrolladores y los responsables de la verificación
- Las personas a cargo de la capacitación de usuarios y escribir manuales de usuarios

- Los clientes
- Los miembros del área comercial o el departamento de ventas que va a posicionar el producto en el mercado.
- El staff legal que elabora contratos de acuerdo a leyes y regulaciones
- Empresas subcontratadas
- Analistas de otros sistemas con los que interactúa

El equipo de desarrollo debe definir el repositorio, el contenido y la estructura del documento.

### **Repositorio**

El repositorio puede ser:

- Software de propósito general como un procesador de textos, una planilla de cálculo, un sistema de manejo de base de datos o una wiki.
- Software específico para administrar requerimientos.

El uso de una herramienta específica permite:

- almacenar atributos descriptivos que se apliquen a varios requerimientos
- administrar los cambios en los requerimientos
- retener versiones históricas
- asignar requerimientos a iteraciones o identificar requerimientos aprobados, derogados o cancelados.
- rastrear requerimientos

Las cualidades de la herramienta que se utilice como repositorio pueden simplificar el trabajo del analista, pero por supuesto no asegura la calidad del modelo. Si el analista no define los términos adecuados en el glosario o no identifica las reglas del negocio que impactan en los requerimientos, la herramienta no supe estas dificultades.

### **Contenido**

El analista debe establecer pautas relacionadas con el contenido:

- Acordar la notación: incluir representaciones visuales, pero no abusar. Si bien en muchas ocasiones se aplica la frase una imagen pueda más que mil palabras, no es un dogma. Un texto en lenguaje natural puede ser también muy expresivo, sobre todo si está estructurado adecuadamente.
- Etiquetar requerimientos para evitar redundancia y para vincularlos y reusarlos.
- Aplicar estilos siguiendo un patrón consistente
- Crear tablas de contenido
- Numerar imágenes y tablas de modo que puedan ser referenciadas.
- Revisar periódicamente el vocabulario y actualizar el glosario

Existen diferentes técnicas para etiquetar requerimientos. Cualquiera sea la elegida cada uno tiene que tener un identificador único y persistente. La más simple es usar un identificador seguido de un número único para el identificador. Al modificar un requerimiento, todos los que estén vinculados a él a través de enlaces, tablas de referencias cruzadas o matrices de trazabilidad, percibirán el cambio. Si un requerimiento se elimina, el número no se reusa.

Por ejemplo CU-1 denota el caso de uso 1, RF-23 denota el requerimiento funcional 23.

Si se usa una herramienta de propósito general es posible vincular requerimiento usando hipervínculos y tablas de referencias cruzadas. Si se usa una herramienta específica en analista debe establecer enlaces para navegar en la información, por ejemplo, cada vez que se menciona una fórmula establecer un enlace que permita visualizar la fórmula.

Esta alternativa es simple pero no permite agrupar ni jerarquizar requerimientos en niveles. Cuando se agrega el requerimiento funcional 156, se le asigna ese número, no queda agrupado con otros que son afines.

Una alternativa es utilizar una jerarquía para identificar requerimientos. El método es simple y compacto. El cuarto requerimiento funcional que aparece en la sección 3.2 del SRS se etiqueta como 3.2.4.

Las herramientas de software suelen soportar esta técnica. Uno de los problemas es que las etiquetas son difíciles de recordar porque suelen tener varios dígitos y no dicen nada del tipo de requerimiento, salvo que uno recuerde qué sección representa cada número. Otro problema es que es difícil modificar la estructura del documento.

Una alternativa intermedia es darle nombre a las grandes secciones y que las letras que identifican a los requerimientos no representen niveles de requerimientos, sino secciones. Por ejemplo, si la sección 3.5 del documento se llama Funcionas de Edición, todos los requerimientos de esta sección se rotulan en secuencia precedidos por el identificador FE, es decir, FE-1 FE-2. Es una organización jerárquica pero las etiquetas son cortas y significativas.

Otra alternativa para organizar etiquetas de manera jerárquica es usar tags. Por ejemplo imprimir.ConfirmarCopias sería el tag para un requerimiento como El sistema solicita confirmación del usuario si se van a imprimir más de 10 copias del documento. El tag Imprimir se aplica a todos los requerimientos ligados con esta actividad. El método es estructurado, expresivo y no lo afectan las inserciones o eliminaciones de requerimientos.

## Estructura

Las plantillas son una alternativa adecuada para organizar el documento en secciones. Existen muchas plantillas diferentes para estructurar el contenido del SRS. Una organización puede adoptar una o usar diferentes plantillas según la envergadura o las características de los productos. Una plantilla posible es:

1. Acerca del proyecto
  - 1.1 Antecedentes
  - 1.2 Objetivos del negocio
  - 1.3 Métricas para el éxito
  - 1.4 Riesgos, premisas y dependencias del negocio
2. Acerca del producto
  - 2.1 Principales características
  - 2.2 Alcance y entrega inicial (lanzamiento)
  - 2.3 Alcance y entregas subsiguientes
  - 2.4 Limitaciones y exclusiones
  - 2.5 Perfiles de los participantes
  - 2.6 Prioridades del proyecto
  - 2.8 Consideraciones del desarrollo
  - 2.8 Consideraciones del desarrollo

Otra posibilidad:

1. Introducción
  - 1.1 Propósito
  - 1.2 Convenciones
  - 1.3 Alcance del proyecto
  - 1.4 Referencias
2. Descripción general
  - 2.1 Perspectiva del producto
  - 2.2 Clases de usuarios
  - 2.3 Entorno operativo
  - 2.4 Restricciones
  - 2.5 Presunciones y Dependencias
3. Características del Sistema
  - 3.x Característica x
    - 3.x.1 Descripción
    - 3.x.2 Requerimientos funcionales
4. Requerimientos para los datos
  - 4.1 Modelo lógico
  - 4.2 Diccionario de datos
  - 4.3 Reportes
  - 4.4 Adquisición de datos, integridad, retención
5. Interfaces externas
  - 5.1 Interfaces de usuarios
  - 5.2 Interfaces de software
  - 5.3 Interfaces de hardware
  - 5.4 Interfaces de comunicación
6. Atributos de calidad
  - 6.1 Usabilidad
  - 6.2 Performance
  - 6.3 Seguridad
  - 6.4 Otros
7. Otros requerimientos

Solo indicar "Otros requerimientos" si son muy diversos, siempre que sea posible establecer un nombre más específico. Una sección que suele ser relevante en muchos proyectos es "Transición" cuando es necesario migrar desde un sistema previo a uno nuevo y es necesario considerar la conversión de los datos y la capacitación de los usuarios.

En cada documento se puede incluir un apéndice con el glosario, que enumera y define los términos específicos que solo se aplican al proyecto. Los términos comunes a varios proyectos es preferible incluirlos en un documento compartido que define los términos específicos de la disciplina que el lector necesita conocer para interpretar correctamente los SRS, incluyendo siglas y abreviaturas. Las definiciones de datos se incluyen en el diccionario de datos no en el glosario.

Con frecuencia una pieza de información puede registrarse en diferentes secciones. Es necesario elegir la que mejor se adecúa y no replicar esa información en cada una de las secciones en las que puede incluirse, a lo sumo agregar referencias cruzadas. También es posible mantener referencias a documentos de otros proyectos, favoreciendo la reusabilidad.

En la descripción de interfaces cada componente se identifica por nombre y versión, se incluye otros sistemas desarrollados por la organización, bases de datos, sistemas operativos, websites y utilitarios. Las condiciones de interoperabilidad se complementan con la sección que describe factores de calidad. La descripción puede incluir los tipos de dispositivos, las interacciones de datos y control y los protocolos usados.

En general los requerimientos de seguridad están vinculados a reglas del negocio, de modo que es importante vincularlos, referenciando al repositorio que especifica políticas o regulaciones de la organización o de su entorno.

En la descripción de interfaces de comunicación es necesario especificar las aplicaciones de correo electrónico, browsers web, protocolos de red, formularios electrónicos. En el desarrollo de sistemas embebidos puede ser necesario generar un documento específico para estos aspectos y referenciarlo desde el SRS.

Cuando no se usa una herramienta de software específica es necesario resolver cómo se va a mantener la historia de las modificaciones sobre el documento y cómo acceder a la última versión.

¿Cuándo terminará?

El resultado de la captura de requerimientos refleja las necesidades de un conjunto de participantes significativos. El resultado del desarrollo de requerimientos define los problemas que el diseñador debe resolver para satisfacer estas necesidades. El diseñador del sistema parte del modelo especificado durante el desarrollo de requerimientos. Si el modelo de diseño que produce puede rastrearse hacia el modelo de análisis que le dio origen, el trabajo del desarrollador se simplifica.

La meta del analista no es elaborar un modelo perfecto, sino capturar los requerimientos que se alinean con los objetivos de la organización, documentarlos y validar el modelo con las partes interesadas, en los tiempos establecidos en el plan del proyecto. El SRS establece un contrato entre el analista y los demás participantes.

Los clientes y usuarios deben estar comprometidos con el proyecto y entender la importancia de enseñarle al analista acerca del entorno. El analista debe aplicar sus capacidades comunicativas y analíticas, aprender acerca del entorno y decidir cuándo el documento de especificación de requerimientos está lo suficientemente completo como para avanzar a la siguiente etapa en el desarrollo del producto.

Es importante evitar llegar a un estado de análisis-parálisis en el cual el desarrollo de requerimientos detenga el desarrollo del producto. No es posible capturar TODOS los requerimientos y tampoco documentar TODAS las necesidades relevadas.

## Tipos de requerimientos

### Requerimientos del proyecto

- Son responsabilidad del adm del proyecto
- Es todo aquello que conduce a las decisiones
- Incluyen:
  - Capacitaciones de los equipos de trabajo
  - Modificaciones sobre el espacio físico
  - Prioridades con relación a otros proyectos
  - Instalación y configuración del producto
  - Transición entre un sistema anterior y el nuevo
  - Adquisición y licenciamiento del producto

- Condiciones para la fabricación, empaquetamiento y comercialización
- Documentación para el usuario, tutoriales, manuales, etc.
- Entrenamiento de los usuarios
- Acuerdos de servicio post venta
- Restricciones de presupuesto
- Restricciones de tiempo
- Migración de datos del sistema actual
- Protección legal y propiedad intelectual

## Requerimientos del sistema

- Son responsabilidad del ing de req
- Son especificaciones de un producto que la organización necesita desarrollar para alcanzar sus objetivos y describen tanto el comportamiento funcional como las cualidades que el usuario considera valiosas, y las restricciones que impone el entorno interno y externo
- Son especificaciones de lo que debe implementarse
- Describen las propiedades internas y externas del sistema, y cómo debe ser su comportamiento
- Algunos requerimientos serán prioritarios o indispensables, otros podrán ser postergables, y otros podrán ser descartados

## Niveles de requerimientos

### Requerimientos del negocio

A partir de las primeras intenciones, que surgen de una necesidad, una oportunidad o un nuevo concepto de producto, el analista y el administrador del proyecto avanzan hasta especificar los requerimientos del negocio que ayudarán a la organización a alcanzar sus objetivos

Están ligados a los requerimientos del proyecto establecidos en el plan del proyecto.

Los requerimientos del negocio son especificaciones de alto nivel, describen los beneficios que la organización aspira a obtener a partir de la construcción del producto: POR QUÉ se construye.

Según el ámbito de desarrollo:

- Inhouse (dentro de una empresa u organización):
  - La necesidad o la oportunidad podría surgir tanto del entorno interno como externo de la organización.
  - El beneficio estará relacionado con cuestiones internas o externas de la organización
  - El cliente será el nivel gerencial, el directorio, el dueño de la empresa, el área de marketing o el departamento de planeamiento de la propia organización
  - En este caso, los requerimientos del negocio sirven de marco para especificar los requerimientos del usuario.
- Empresa comercial:
  - Los requerimientos del negocio surgen de una oportunidad del mercado o de una idea de producto.
  - El beneficio estará relacionado con la comercialización del producto.

- El cliente será el área de comercialización que tendrá a su cargo la venta del producto cuando esté desarrollado
- Cuando el área de desarrollo y el área comercial acuerdan los requerimientos del negocio, el analista podrá avanzar hacia los requerimientos funcionales surgidos de los requerimientos del negocio, y los mismos serán validados con los usuarios o clientes potenciales del producto

Ayudan a decidir acerca de la terminación de la etapa de desarrollo de requerimientos.

Un proyecto que no define requerimientos del negocio, o no los especifica con la suficiente precisión, corre un riesgo importante de fracaso: los participantes pueden no compartir un objetivo común y tener visiones diferentes o incluso opuestas respecto al producto final. De ser así, el producto puede tener que ser modificado varias veces hasta la entrega final, sin cumplir los tiempos establecidos en el plan del proyecto.

Los requerimientos del negocio incluyen:

### Objetivos del negocio

Los objetivos del negocio permiten analizar y medir los resultados. Cada objetivo debe formularse cuantificando el beneficio de modo que pueda ser medido, y debe relacionarse con una métrica del éxito, que puede provenir de patrocinadores, ejecutivos de la organización o gerentes.

El analista define los objetivos del negocio junto con el administrador del proyecto, o bien, en base a los objetivos del proyecto definidos previamente por el administrador.

Los objetivos del proyecto expresan el beneficio que se espera alcanzar a partir de los siguientes interrogantes: ¿por qué y para qué se desarrolla el producto?. Los objetivos del negocio cuantifican este beneficio. El analista y el administrador, definirán en conjunto cómo medir estos resultados.

### Visión del producto

Permite decidir cuándo se completó el proyecto, y favorece la toma de decisiones cuando surgen cambios en los requerimientos y propuestas de mejoras y extensiones

Brinda el contexto para la toma de decisiones durante el desarrollo del producto y alinea a todos los participantes.

La visión del producto es estable y se aplica al producto como un todo.

Especifica:

- Para quienes se desarrolla el producto
- El problema que le da origen
- En qué categoría se puede incluir
- Cuáles son las principales cualidades y características que lo distinguen
- Qué ventajas provoca su utilización

Plantilla para especificar la visión del producto

PARA [cliente que demanda el producto]

QUIEN [necesidad, oportunidad o idea]

EL [nombre del producto]

ES [categoría del producto o tipo de proyecto]

QUE [características y cualidades distintivas]

EN LUGAR DE [sistema o proceso actual que se va a reemplazar, o producto de la competencia]

EL NUEVO PRODUCTO [ventajas competitivas o principales beneficios]

## Alcance del proyecto

La especificación del alcance ayuda a resolver discusiones sobre las características que el sistema debería o no debería exhibir, y las funcionalidades que deben o no deben estar incluidas en el producto final.

Establece los límites entre lo que queda dentro y fuera de la iteración actual dentro del proceso de desarrollo del sistema.

Es dinámico. La meta del equipo de desarrollo es administrar el alcance de un proyecto específico definido como etapa parcial de la visión del producto final. Los participantes ajustan el alcance de acuerdo a los recursos, los requerimientos de calidad, las restricciones, siempre en el marco de la visión del producto.

Los objetivos del negocio, la visión del producto y el alcance del proyecto, se especifican en base a las características más importantes, y a los principales flujos de datos que comunican al sistema con los actores con los que interactúa.

En la documentación de requerimientos del negocio, se incluyen también otros temas relevantes, aunque no sean requerimientos:

- Limitaciones
- Riesgos
- Partes interesadas
- Clases de participantes que son fuente de requerimientos

## Documento de visión y alcance

Establece los requerimientos del negocio a partir de los cuáles se desarrollará el resto del proyecto.

Opiniones según autores:

1. El documento de visión y alcance define aspectos de alto nivel, los detalles se especifican en otros documentos. En cada iteración de ciclo de vida del sistema, se inicia un proyecto de mejora o ampliación del producto que tendrá su propio documento de requerimientos, tomando como marco de referencia el documento de visión y alcance.
2. Es preferible mantener un único documento de especificación de requerimientos, que va a ir evolucionando a lo largo del desarrollo del sistema. Será una especie de bitácora en donde se acumule el historial de todo el desarrollo del producto.
3. Cada iteración del ciclo de vida del desarrollo del sistema tendrá su propio documento de especificación de requerimientos que incluirá una sección para la visión y alcance. El lanzamiento de una nueva versión o mejora del producto tendrá su propio documento.

## Plantilla de documento de visión y alcance

### Requerimientos del negocio

- Antecedentes: Describen el sistema, manual o automático, que se está usando actualmente, los circuitos y procesos, sus limitaciones y los obstáculos que plantea.

Resumen la historia o la situación a partir de la cual se tomó la decisión de desarrollar un producto.

- Necesidades del negocio, oportunidad del mercado o idea del producto:
  - Cuando el sistema se desarrolla dentro de la misma empresa, describe el problema que debe ser resuelto o el proceso que debe mejorar, así como también el entorno en el cual se usará.
  - Cuando el sistema se desarrolla en una empresa comercial, se describe la oportunidad que se percibe en el mercado y los productos contra los que competirá, y las ideas centrales que le dan origen.
- Objetivos del negocio: Los objetivos están claramente ligados a las necesidades, oportunidades e ideas del negocio. Por cada necesidad, oportunidad e idea puede plantearse un objetivo cuantificable. pueden ser financieros o no. Pueden estar cuantificados directamente o acompañados por una métrica que indique cómo se van a evaluar los resultados.
- Visión del producto: Especifica el destinatario y tipo de producto, sus características y cualidades, las ventajas comparativas.
  - Cuando el ámbito de desarrollo es una empresa comercial, la dificultad es reflejar los intereses y expectativas de distintos clientes, unos priorizando las limitaciones de recursos, otros enfocados en la política estratégica corporativa, otros guiados por la situación del mercado.
  - Muestra cómo alinear las tendencias del mercado, la innovación tecnológica, las estrategias corporativas.
  - Enumera tecnologías, procesos y recursos requeridos para brindar una solución completa.
  - Describe las necesidades de un cliente típico dentro del mercado al que se orienta el producto.
- Métricas para el éxito: indican si un proyecto está encarrilado para alcanzar sus objetivos del negocio. Especifican cómo van a medir los clientes y usuarios el éxito del proyecto.

El administrador del proyecto es responsable de identificar los factores que tendrán mayor impacto sobre el éxito, incluyendo aquello que la organización no puede controlar.

El analista es responsable de seleccionar los participantes, favorecer la comunicación y colaborar con la especificación.

Durante la captura se consulta cómo se puede decidir si el objetivo se satisfizo o si el problema está resuelto.

Se debe definir cuándo se esperan alcanzar los resultados.

En ocasiones los objetivos del negocio no pueden medirse hasta que el proyecto termina. En otros casos existe una dependencia entre los objetivos del negocio de este proyecto y otros previos. Algunas métricas se pueden rastrear durante las pruebas, otras poco después del lanzamiento y en otros casos puede pasar mucho tiempo hasta notar el impacto.
- Riesgos del negocio: Resumen los principales acontecimientos que representan amenazas potenciales para el negocio, asociados con desarrollar o no el producto. Incluyen:
  - competencia en el mercado,
  - aceptación del usuario,
  - instalación

- impacto negativo en el negocio.
- NO son los riesgos del proyecto, que están determinados por los recursos y factores tecnológicos.
- Premisas del negocio y dependencias: Una premisa es una frase que se supone verdadera en ausencia de una certeza que indique lo contrario. El analista debe explicitar las premisas de cada uno de los participantes para evitar inconsistencias. Registrar todos los supuestos que dependen de factores externos y que el administrador del proyecto debe monitorear regularmente.

### **Alcance y limitaciones del proyecto**

- Principales características: El alcance del proyecto incluye las principales características y capacidades, enfatizando aquellas que se agregan respecto a versiones previas o productos competidores. Cada característica debe estar etiquetada para permitir hacer un seguimiento, si es necesario agruparlas en categorías.
- Alcance y entrega inicial:
  - Identificar las características y capacidades que se incluirán en la primera entrega del producto.
  - Justificar por qué se eligen para la primera entrega y qué tipos de usuarios se benefician con estas características y capacidades.
  - Describir las características de calidad y beneficios para cada clase de usuario de esta primera entrega.
  - Evitar incluir todas las características y capacidades en el primer entregable, de modo que sea factible cumplir con el plan.
- Alcance y entregas subsiguientes: en cada etapa los entregables agregan funcionalidades o características, o enriquecen las capacidades iniciales.
- Limitaciones y exclusiones: El alcance de cada proyecto queda definido en alto nivel, alineados con los objetivos del negocio y la visión del producto. El producto brindará algunas funcionalidades y características y NO ofrecerá otras. Es importante establecer tanto lo que el sistema hace y como lo que no hace en cada etapa y al completarse el proyecto.
- Aspectos asociados al proyecto:
  - Enumerar toda la información y las actividades que va a demandar el desarrollo para alcanzar una solución efectiva en el entorno operativo
  - Describir los cambios necesarios en el acceso a redes, almacenamiento de datos, migración de datos.
  - Indicar la capacitación y entrenamiento de usuarios que se requiere y las modificaciones en el proceso.

### **Entorno del producto**

- Perfiles de los participantes: Los participantes son las personas, grupos o bien organizaciones que pueden verse afectadas por el sistema o lo afectan ellas mismas a él. Los perfiles enumeran a todas las clases de participantes y describen a las más importantes, en especial a los usuarios directos. Al especificar el perfil de los participantes, es importante registrar:
  - El principal beneficio que espera cada cliente., que restricciones percibe cada uno

- La actitud de cada clase de usuarios hacia el producto y hacia los cambios en los procesos operativos.
- Las restricciones y características que son valiosas para algunos usuarios en particular.
- Qué características le interesan a cada clase de usuario, cómo van a afectar su productividad, cuánto se reduce el esfuerzo, qué atributos de calidad consideran valiosos.
- Los errores que se buscan prevenir en las tareas que realizan algunos usuarios en específico
- Los estándares o regulaciones que se deben cumplir
- Prioridades del proyecto: Acuerdo de prioridades entre las partes interesadas, a fin de reducir conflictos en la toma de decisiones. Se deben considerar 5 dimensiones:
  - Características
  - Calidad
  - Tiempo
  - Costo
  - Equipo de desarrollo

Por cada dimensión se deben considerar las restricciones y los grados de libertad a la hora de tomar decisiones:

- Restricciones: límites dentro de los cuáles debe operar el administrador del proyecto
- Grados de libertad: factores que el administrador del proyecto puede ajustar.

A lo largo del desarrollo de requerimientos, los requerimientos del negocio van a permitir planificar la captura de las reglas del negocio, y de cada uno de los niveles de los requerimientos que siguen; definir prioridades y evaluar pertinencia (cualquier requerimiento del usuario que no contribuya a alcanzar el objetivo del negocio o quede fuera del alcance del proyecto, no debería considerarse en el documento de especificación de requerimientos, o bien exige revisar los requerimientos del negocio).

### Reglas del negocio

Son las políticas, normas y convenciones de la organización, los estándares de la industria y las regulaciones del gobierno que conducen algún aspecto del negocio. Las normas internas no pueden relajar a las normas externas

Si bien no son requerimientos del producto, dan origen a muchos requerimientos. Son guías que conducen las acciones y procedimientos. Pueden derivar restricciones o funcionalidades.

Los procesos de negocio son todas las actividades que realiza una organización para alcanzar sus objetivos.

Los requerimientos del negocio son las razones que justifican el desarrollo de un producto con el cual se pretende mejorar alguno de los procesos de negocio para alcanzar un beneficio.

Los sistemas de información suelen automatizar algunos procesos de negocio aportando eficiencia y otros beneficios alineados con los objetivos del negocio.

Las reglas del negocio imponen restricciones, disparan acciones y gobiernan los procesos. No son en sí mismas requerimientos de software, sin embargo son fuentes de requerimientos porque evitan propiedades que el sistema debe cumplir.

Toda organización trabaja de acuerdo a un conjunto de políticas, leyes y estándares.

Las reglas del negocio están conformadas tanto por la reglamentación externa e interna, las normas formales y las convenciones.

Las mismas reglas pueden aplicarse a distintos procesos.

Una nueva norma puede ser el origen de un proyecto para desarrollar un nuevo sistema o modificar uno existente.

Un nuevo sistema que modifique procedimientos administrativos o procesos de producción, puede requerir modificar la normativa para adecuarla a los cambios.

La organización puede modificar sus convenciones o normas internas pero no las normas externas.

### **Captura de reglas de negocio**

Las reglas pueden estar asociadas a normas estrictas o a convenciones.

Las normas estrictas son más fáciles de capturar porque están documentadas, y en este caso el análisis de documentos es la técnica más adecuada: leyes, estándares de la industria, políticas corporativas, contratos, planes de negocio.

Cuando se trata de un proyecto de reemplazo, se pueden deducir las reglas relevantes analizando el código o la documentación del sistema actual, para comenzar a identificar normas estrictas y convenciones.

Los permisos y niveles de privacidad de los datos permiten identificar restricciones para los roles.

Las entrevistas también serán adecuadas para capturar tanto convenciones como normas estrictas.

EN TODOS LOS CASOS ES NECESARIO CONSIDERAR LA PERTINENCIA Y LA VIGENCIA DE LAS REGLAS CAPTURADAS.

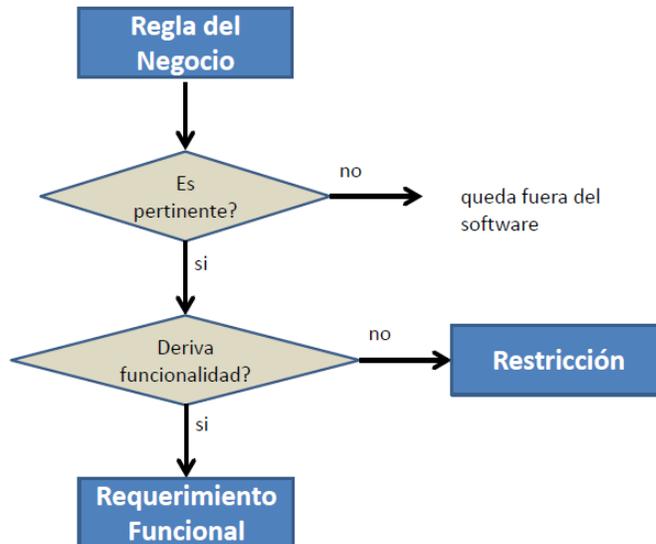
### **Análisis de las reglas de negocio**

Después de identificar las reglas del negocio, el analista decide cuáles afectan al desarrollo de software.

El analista deriva requerimientos funcionales y para los datos, a partir de las reglas del negocio, en particular, a partir de las restricciones

Durante la etapa de análisis el ingeniero de requerimientos define:

- qué reglas no afectan al sistema
- qué reglas afectan a los procesos pero no al software
- qué reglas derivan funcionalidad
- qué reglas restringen funcionalidad



Con frecuencia los clientes y usuarios conocen las reglas pero saben también que se aplican parcialmente.

Al automatizar los procesos el margen de tolerancia desaparece y es necesario definir el nivel de control que se va a aplicar.

Con frecuencia se definen permisos que habilitan *excepciones* para algunos usuarios.

## Especificación de las reglas del negocio

Las reglas del negocio deben documentarse de modo que puedan usarse en diferentes desarrollos.

A cada regla del negocio se le asigna un identificador único de modo que puedan ser referenciados desde otros documentos o representaciones visuales.

Las referencias abrevian los modelos y aseguran cierto nivel de consistencia. Además de asegurar la consistencia, la documentación de las reglas del negocio favorece el reuso.

### Reglas atómicas y compuestas

Una regla atómica es indivisible.

Una regla compuesta combina dos o más reglas atómicas o compuestas.

El análisis de consistencia y la derivación de requerimientos funcionales se simplifica cuando las reglas son atómicas.

## Tipos de repositorio

- un catálogo simple
- una base de datos
- una herramienta colaborativa como una wiki
- un software específico para la gestión de requerimientos

Es importante etiquetarlas adecuadamente para que puedan referenciarse desde el documento de especificación de requerimientos, así las modificaciones en las reglas no requieren cambios en otros documentos o representaciones visuales.

Cualquiera sea el repositorio, lo ideal sería poder consultar las reglas del negocio de acuerdo a diferentes criterios:

- Palabras claves
- Palabras en el texto
- Órgano de emisión
- Fecha

También es importante mantener un registro de las reglas que ya no tienen vigencia, pero que en algún momento afectaron al producto.

## **Trazabilidad**

Así como es importante asignar una identificación a cada regla que permita referenciarla, también es importante vincular a cada requerimiento funcional con la regla que lo origina, manteniendo una referencia al indicador.

Una matriz de trazabilidad es una herramienta que conecta requerimientos funcionales con reglas de negocio.

Si las reglas del negocio y los requerimientos se guardan en archivos convencionales pueden conectarse usando hipervínculos.

## **Motores de reglas de negocio**

Las reglas del negocio son tan importantes que en los últimos años se han desarrollado diversas iniciativas para estandarizarlas y automatizarlas a través de motores de reglas de negocio, tomándolas como requerimientos, siempre y cuando cumplan con ciertas condiciones:

- Se deben expresar de manera tal que su exactitud o precisión pueda ser validada por expertos
- Se deben expresar de manera tal que pueda verificarse su coherencia
- Se deben expresar utilizando lógicas formales, como la lógica de predicados

## **Clasificación de las reglas de negocio (DR14)**

→ **Hechos:** Un hecho es una frase que es verdadera en un momento determinado de tiempo. Un hecho describe asociaciones entre términos importantes del negocio. No todos los hechos capturados van a ser relevantes, pero el analista debe analizar cuidadosamente cuáles descartar. Los hechos acerca de entidades relevantes del sistema aparecen en el modelo de datos.

→ **Restricciones:** Una restricción es una frase que limita las operaciones o funcionalidades que el sistema o los usuarios pueden realizar. Algunos autores consideran que una restricción no sólo establece las acciones que no pueden o no deben realizarse, o qué solo algunos roles puede realizar, sino que también determina las operaciones o funcionalidades que deben realizarse.

- ◆ **Matriz de roles y permisos:** La matriz de roles y permisos es un modelo que define los tipos de roles y sus permisos asociados necesarios para ejecutar operaciones en el sistema.
  - Una operación puede ser una funcionalidad individual en el sistema o un grupo de funcionalidades y puede ser conceptual o un elemento físico real en la interfaz de usuario.
  - Un rol es un nombre para una colección de usuarios que comparten operaciones comunes y acceso a un sistema. Un rol puede corresponder a un cargo en la organización o un área completa. Puede ocurrir que con el mismo cargo los usuarios ocupen distintos roles y con distintos cargos tengan asignado el mismo rol.
  - Los permisos PUEDEN considerar las convenciones de la organización pero DEBEN modelar las normas referidas a privacidad de los datos y otras restricciones legales. Los permisos pueden estar asociados a las operaciones o funcionalidades, a los datos o bien a las áreas de la organización.
- **Disparadores de Acciones:** Un evento disparador de acciones es una regla que puede iniciar alguna actividad específica bajo determinadas condiciones.
  - ◆ **Tabla de decisiones:** Una tabla de decisión permite modelar acciones cuya ejecución depende de una combinación de condiciones. Si las condiciones deben evaluarse en un orden determinado es preferible usar un árbol de decisión. Las ventajas de una tabla de decisión sobre un árbol de decisión son la trazabilidad visual y la simplicidad para construirla.
- **Inferencias:** Una inferencia es un hecho que se deriva de otro hecho. Suelen tener la misma forma condicional que los disparadores de acciones, pero la consecuencia no es una acción sino un hecho.
- **Reglas de Cómputo:** Un cómputo es una regla que indica cómo transformar un dato dado en un nuevo dato usando una fórmula o algoritmos. Algunas reglas de cómputo están reguladas por el entorno externo a la organización.

## Captura de requerimientos del negocio

Los clientes probablemente no formulen sus necesidades, ideas u oportunidades de manera clara, ordenada, precisa, consistente. Con frecuencia intercalan información irrelevante o con requerimientos que no corresponden al producto. El analista debe aclarar inconsistencias, imprecisiones y validar los requerimientos especificados.

Una vez especificados y validados los requerimientos del negocio, cada encuentro de captura debería iniciarse con la lectura de los objetivos de negocio y la visión del producto.

Terminar cada encuentro organizando los requerimientos teniendo en cuenta las prioridades con respecto a los objetivos y la pertinencia con relación al alcance del proyecto.

## Técnicas de representación visual para modelar requerimientos de negocios

Las representaciones visuales complementan la especificación del documento de visión y alcance. El propósito es facilitar la comunicación con los participantes, a partir de modelos gráficos. Cada alternativa puede resultar más o menos adecuada dependiendo del tipo de proyecto.

Calidad de la representación visual:

Un diagrama es CORRECTO si refleja:

- Los requerimientos capturados por el analista
- Las necesidades reales del usuario

Un diagrama es ÚTIL si permite:

- Validar los requerimientos expresados por clientes y usuarios
- Visualizar los requerimientos especificados por el analista

## Árbol de características

Es un modelo que organiza las características en grupos lógicos mostrando el alcance completo de una solución en una única representación gráfica.

Es un diagrama de alto nivel. Modela el alcance del proyecto a partir de grupos de características organizadas en forma jerárquica.

Es ideal para validar el alcance del proyecto y acordar sus limitaciones.

Permite identificar ausencias, malentendidos, premisas erróneas, redundancias, huecos.

Anticipa la descomposición funcional de las características.

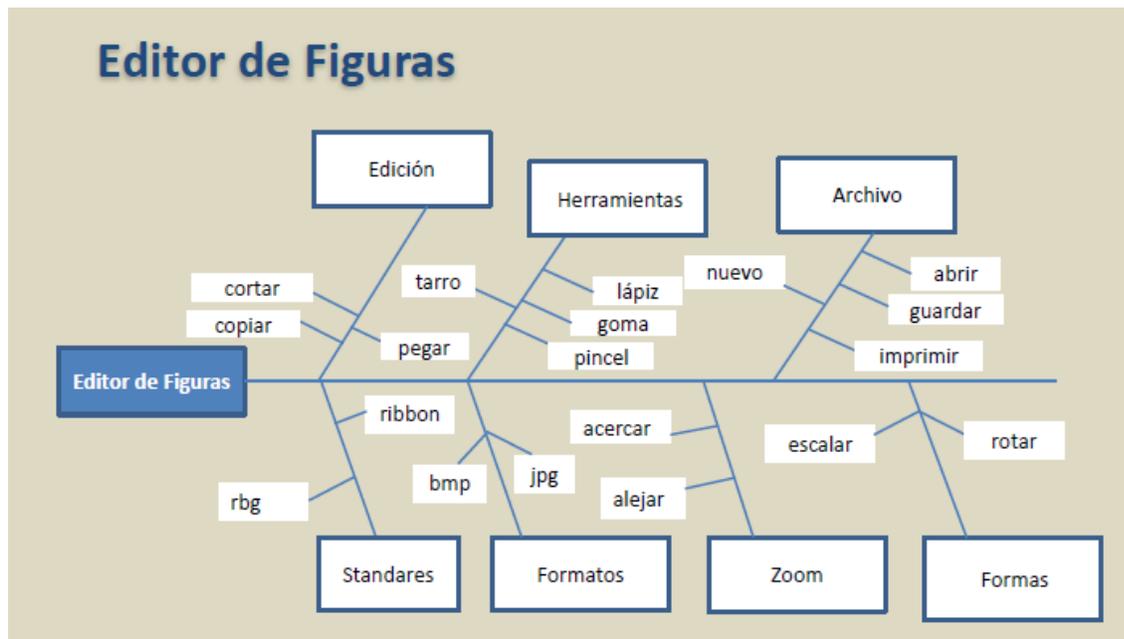
En el árbol, cada rama representa una característica, y cada subrama es también una característica pero más específica.

Por lo general se organiza con 2 o 3 niveles (L1,L2,L3).

El nivel 0 (L0) es el tronco del árbol, y en el extremo se nombra al sistema global.

El nivel 1 (L1) agrupa un conjunto de características relacionadas con algún criterio.

NO SE MODELAN LAS FUNCIONALIDADES INDIVIDUALES.



## Creación del árbol:

1. Armar la lista de características
  - a. Una característica concentra funcionalidades o cualidades valiosas para alcanzar los objetivos del negocio.
  - b. Debe haber coherencia en el uso de sustantivos y verbos y también en su conjugación (uniformar la notación)
  - c. Las características se representan como frases verbales breves, no más de 3 palabras, en el extremo de una rama.
  - d. Cada característica debe representar a una colección de requerimientos relacionados entre sí con algún criterio lógico. No es la intención reconocer requerimientos individuales ni funcionalidades específicas.
2. Agruparlas en categorías (características de alto nivel, agrupadas de acuerdo a un criterio más general)
  - a. Selección de criterio para agruparlas: tener en cuenta clases de participantes, niveles de acceso, reportes
    - i. Por objetivos del negocio
    - ii. Por subproyectos o entregables
    - iii. Por roles o clases de usuarios
  - b. Separar características por niveles de especificidad de acuerdo al criterio elegido
  - c. Identificar características específicas "sin padre" que no correspondan a ninguna de las características generales identificadas
  - d. Identificar características generales "sin hijos"
  - e. Identificar redundancias (una misma característica se repite en varias categorías)
  - f. Decidir cantidad de niveles y nivel de granularidad (nivel de detalle con el que se va a modelar las características)
3. Dibujar el árbol (no más de 3 niveles, ni más de 8 categorías en cada nivel): se dibuja cuando ya se han definido los niveles y distribuido las características entre niveles. Como todo árbol es posible construirlo también desde el tronco hacia las ramas inferiores. Es un proceso dinámico, se pueden agregar características luego de que ya se han armado las categorías y también es posible reagrupar y definir nuevas categorías.
4. Validar el árbol: se pueden identificar ramas superpobladas o poco pobladas.

## Aplicaciones del árbol de características

El árbol de características se construye durante las primeras iteraciones, pero es utilizado durante todo el proceso de desarrollo de requerimientos para contextualizar cada una de las necesidades formuladas por los usuarios, organizar la captura de requerimientos de usuario, derivar requerimientos funcionales, resolver conflictos, establecer prioridades y gestionar cambios.

Cuando se reemplaza un sistema, el árbol de características del sistema a reemplazar es valioso para evaluar la envergadura de la mejora o ampliación.

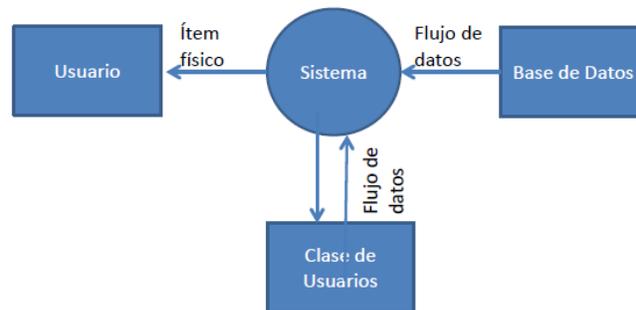
Cuando el proyecto es complejo y el equipo de desarrollo es numeroso, puede servir para distribuir el trabajo.

También es útil para comparar diferentes productos, siempre que ambos árboles estén elaborados con el mismo nivel de granularidad.

## Diagrama de contexto

Permite visualizar el alcance del proyecto, los límites y conexiones entre el sistema y su entorno. El objetivo es describir el flujo de datos e ítems físicos que intercambia con entidades de su entorno. Modela a las entidades externas o terminales y el flujo de datos o ítems físicos entre estas terminales y el sistema.

Los detalles de los datos y procesos no son visibles.



### Clasificación de las entidades externas

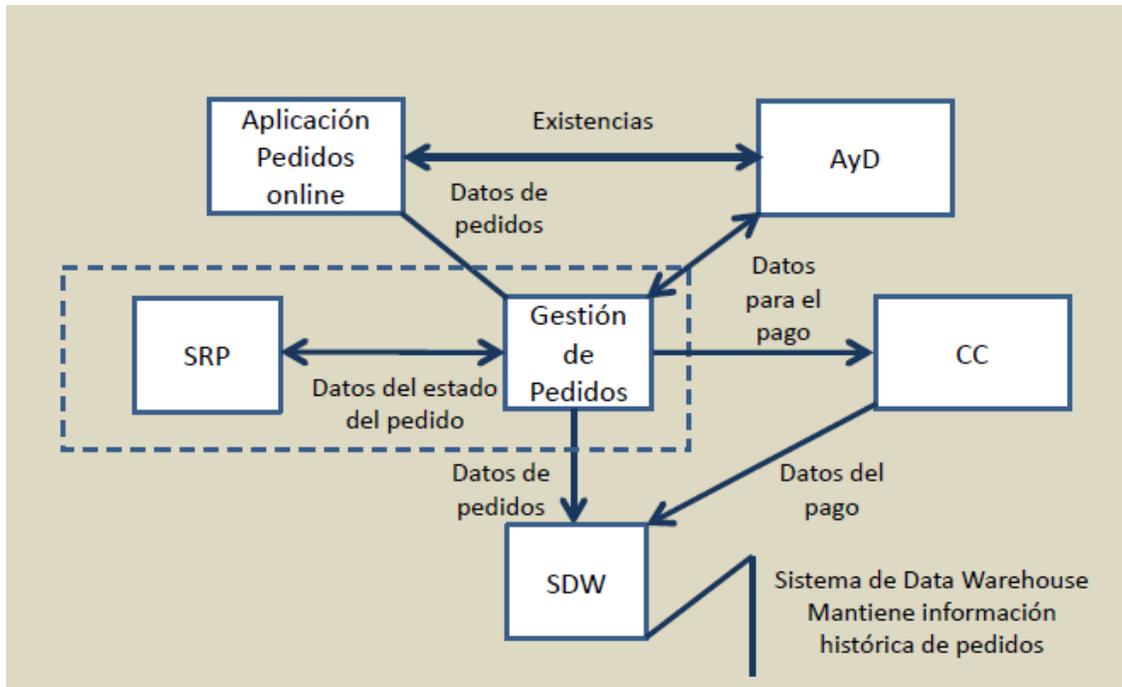
- Activas: son dinámicas y frecuentemente inician eventos para lograr algún objetivo o propósito
- Pasivas: son estáticas y no interactúan frecuentemente con el sistema
- Cooperativas: son predecibles y pueden ser utilizadas por el sistema para alcanzar algún resultado deseado
- Autónomas: están separadas del sistema pero lo afectan indirectamente imponiendo restricciones o influencias similares.

### Creación de un diagrama de contexto

1. Identificar entidades externas, actores o terminales que son fuentes o destinatarios de los datos o ítems físicos
2. Identificar flujos de datos entre actores y sistema
3. Dibujar el diagrama

## Mapa de ecosistema

Modela a un conjunto de sistemas relacionados como un ecosistema de soluciones, para sistematizar el análisis de las relaciones entre los sistemas y garantizar que han sido considerados todos los sistemas que interactúan para solucionar el problema. Pueden construirse en un encuentro de captura.



### Creación del mapa de ecosistema

1. Identificar sistemas
2. Identificar interfaces
3. Construir el mapa

Los sistemas se representan con rectángulos.

Las interfaces entre sistemas se representan con líneas que conectan los rectángulos.

La dirección de los datos se indican con flechas que marcan la dirección en la que fluyen los datos entre los sistemas. Se pueden etiquetar.

Los subsistemas se pueden representar agrupando varios sistemas dentro de un recuadro de líneas punteadas.

La información suplementaria puede agregarse con llamadas con texto.

Debe complementarse con especificaciones que describan los detalles de la comunicación o interfaces entre sistemas.

Herramientas para la captura, para identificar sistemas e interfaces:

- Diagramas de flujo de datos
- Diagramas de datos
- Flujos de procesos

Beneficios del mapa de ecosistemas

- Identificar flujos de procesos y partes interesadas
- Capturar requerimientos de los datos (entrada y salida)
- Capturar atributos de calidad y restricciones (seguridad e interoperabilidad)
- Evaluar impacto de los cambios sobre los sistemas relacionados

Para que NO SIRVE:

- NO modela secuencias de interacciones entre sistemas ni el flujo detallado de datos ni tampoco los eventos o decisiones
- NO es útil cuando el sistema es completamente autónomo, como el caso de los empaquetados.

Luego de que se hayan especificado los requerimientos del negocio, el equipo de desarrollo queda comprometido a construir un sistema que satisface la visión del producto, con el alcance especificado, en el tiempo estimado, y con los recursos asignados en el plan del proyecto.

## Requerimientos del usuario

Especifican las tareas que los usuarios esperan realizar con el sistema. El analista interpreta los requerimientos del usuario y deriva la funcionalidad que deberá implementarse.

En este enfoque van a surgir aquellas características que serán relevantes, que se utilicen más frecuentemente, y aquellas que no sean tan frecuentes, e incluso que no se utilicen.

Clasificación:

- Indispensables
- Postergables u opcionales
- Hipotéticos

Los requerimientos de baja prioridad pueden estar relacionados con futuros cambios, y el considerarlos anticipadamente puede reducir el impacto.

Cada requerimiento de usuario define un tipo de tarea que un tipo de usuario específico realizará con el sistema, o una característica que un tipo específico de usuario desea del sistema.

Cuando el producto se desarrolla para un área interna de la organización, el desarrollo de requerimientos del usuario implicará conocer los perfiles de los usuarios:

- Antecedentes
- Formación
- Conocimientos y habilidades
- Motivaciones
- Obstáculos y necesidades
- Actitudes y conflictos
- Tareas y procesos en los que participa
- Responsabilidades

## Atributos de calidad

Los usuarios manifiestan los atributos o cualidades que esperan que el sistema exhiba para satisfacer sus necesidades.

## Restricciones

Los usuarios pueden plantear restricciones relacionadas a las interfaces, a aspectos económicos o aspectos operativos.

Son requerimientos que limitan las opciones disponibles para los desarrolladores, con relación a la construcción del sistema.

La especificación de atributos y restricciones está ligada a las funcionalidades del sistema y también al entorno interno y externo.

## Técnicas de especificación para requerimientos de usuario

### Escenarios

Modela un conjunto de interacciones del usuario con el sistema mientras realiza una tarea concreta. Brindan ejemplos realistas y detallados de cómo los usuarios podrán llevar a cabo sus tareas en el sistema.

El objetivo principal es proporcionar ejemplos del uso futuro del nuevo producto como ayuda para comprender y aclarar los requerimientos del usuario y proporcionar una base para los test de aceptación posteriores.

Pueden ayudar a identificar objetivos de usabilidad y tiempos probables para la terminación de las tareas.

Promueve la aceptación del desarrollador y alienta un enfoque de diseño centrado en el usuario.

Es una instancia específica de un caso de uso.

### Modelos de casos de uso

Se utiliza para especificar las interacciones del usuario con el sistema mientras realiza sus tareas.

Incluye un conjunto de diagramas de caso de uso y un conjunto de especificaciones breves y detalladas de los casos de uso.

Se crea durante el desarrollo de requerimientos, guía todo el proceso de desarrollo de un sistema de información y se utiliza durante todo el ciclo de vida.

Describe una tarea discreta e independiente que un actor realiza para lograr algún resultado valioso. Abarca un conjunto de actividades relacionadas que conforman una tarea.

## Diagramas de casos de uso

Un diagrama de casos de uso es una representación visual de alto nivel que modela un conjunto de interacciones entre el sistema y su entorno (uno o más actores mientras usan el sistema).

Es particularmente útil cuando el analista se concentra en las tareas que realiza el usuario, aunque también se usa cuando el foco está en las funcionalidades del producto.

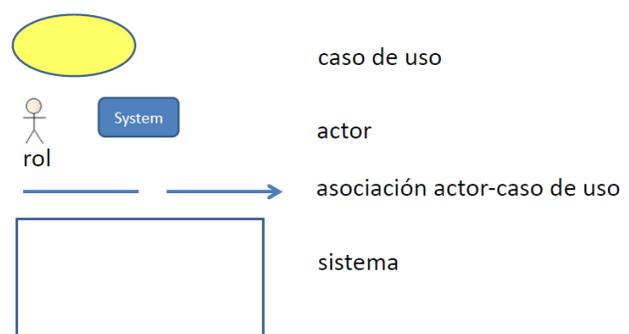
Cada caso de uso describe un diálogo particular entre el sistema y al menos un actor del entorno.

Para cada interacción es posible establecer con precisión el comienzo y la terminación.

Consta de 4 componentes:

- Caso de uso: representadas mediante elipses rotuladas con el nombre del caso

### Diagrama de Casos de Uso



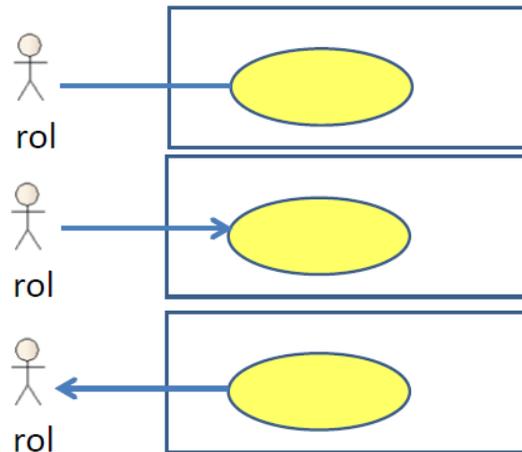
- Actores: pueden representar componentes de hw, personas, o a otros sistemas que incluyen componentes de sw, representados por una persona con su rol o un rectángulo con el nombre del sistema o el componente
- Asociaciones: entre actores y caso de uso representadas por líneas o flechas
- Sistema: (límites del sistema que los separa del exterior), representado por un rectángulo.

En las asociaciones entre actores y casos de uso, las representaciones pueden ser líneas o flechas.

La asociación indica que el actor usa, ejecuta o interactúa o dialoga con el caso de uso.

Una flecha que va desde un actor a un caso de uso, indica que el actor ocupa un rol principal, inicia el caso de uso.

Una flecha que va desde el caso de uso al actor, indica que el actor ocupa un rol secundario, es decir, que tiene una participación en el caso de uso, pero no lo inicia.



## Definiciones

### Caso de uso

- Un **caso de uso** representa a una tarea que realiza un **actor** con el sistema y que permite obtener un resultado.
- Un **caso de uso** representa a una interacción o diálogo entre un **actor** y el sistema, que permite obtener un resultado.
- Un **caso de uso** representa a un uso entre un **actor** y el sistema, que permite obtener un resultado

### OJO!! A TENER EN CUENTA

- Cada caso de uso debe representar a una unidad de tarea, no a cada una de las acciones que la conforman.
- Un caso de uso puede ser una acción simple, porque la tarea del usuario es una acción simple y nos interesa que quede modelada.
- No se modelan como casos de uso las acciones que computa el sistema sin interactuar con el entorno.
- No se modelan las tareas que realiza el usuario como parte del proceso pero sin interactuar con el sistema.
- No se modelan piezas de diseño, por lo tanto no hay alusión a la interfaz de usuario.
- La primera palabra del nombre del caso de uso debe ser un verbo en infinitivo.
- Cada término tiene que ser claro, significativo y con una interpretación precisa. Todas las partes interesadas tienen que interpretar de la misma manera a los términos específicos del dominio.

## Actor

Un **actor** representa un **rol** que puede estar asignado a una persona, un dispositivo de hardware o un sistema de software que interactúa con el sistema para ejecutar un caso de uso.

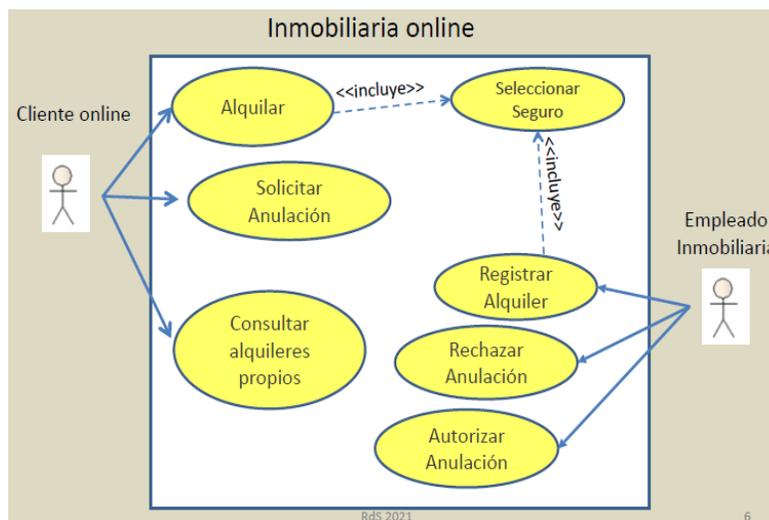
Los actores NO forman parte del sistema sino de su entorno, son entidades externas.

Distintas entidades pueden ocupar el mismo rol, y una misma entidad puede ocupar distintos roles.

En el diagrama NO se modelan las interacciones entre actores.

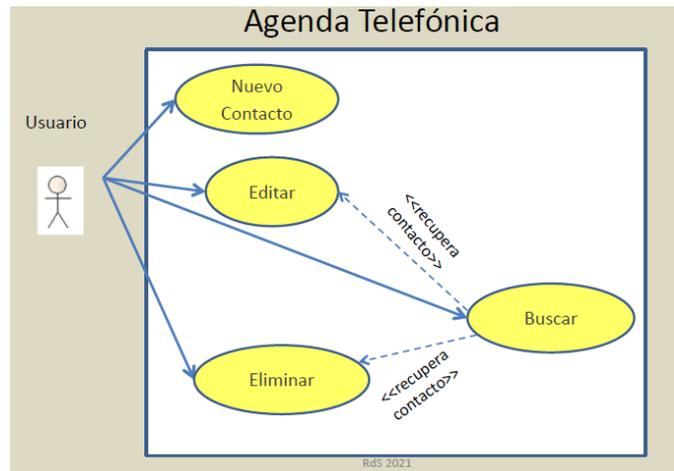
## Relaciones entre casos de uso en un DCU

- Inclusión
  - Un caso de uso base puede "incluir" otro caso de uso.
  - El resultado del caso base depende del resultado del caso de uso incluido o subcaso.
  - Decimos que el caso base incluye al caso incluido.
  - El caso incluido es parte esencial del caso base.
  - El caso base no puede completarse si no se realiza el caso incluido.
  - El caso base y el caso incluido modelan una tarea que el usuario realiza en una misma sesión.

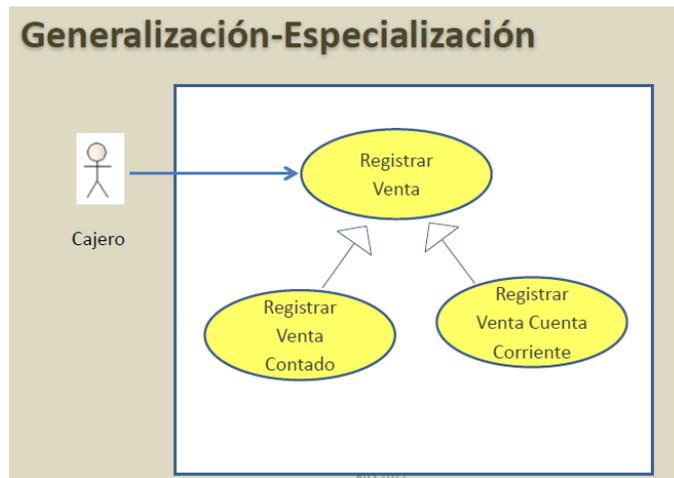


- Extensión
  - Un caso de uso base puede "extenderse" en otro caso de uso.
  - El resultado del caso general puede depender del resultado del caso de uso que lo extiende.
  - La vinculación se rotula con un término significativo
  - Decimos que el caso extendido extiende al caso base.
  - El caso extendido NO es indispensable para el caso base. El caso base puede completarse aún si no se completa el caso de uso que lo extiende.
  - Permite modelar que una tarea se produce siempre y otra se agrega a la primera en algunos casos. Sólo se ejecuta bajo ciertas condiciones o circunstancias.

- Algunos autores consideran que el caso extendido permite modelar el flujo anormal



- Generalización-Especialización
  - Un caso de uso general se especializa en dos o más casos específicos.
  - Los actores pueden estar ligados tanto al caso general como a los casos específicos.



## Relaciones entre actores en un DCU

Un actor representa un rol que va a ser ocupado por alguna entidad externa al sistema para un uso específico.

El conjunto de actores pueden estar vinculados por una relación de generalización - especialización.

Si el actor E es una especialización del actor G, E interactúa con los mismos casos de uso con los que interactúa G.



## Construcción del DCU

La construcción del modelo es iterativa, partiendo de casos muy generales y agregando luego otros más específicos o agregando extensiones o inclusiones.

Algunos equipos de trabajo no sólo especifican las interacciones, sino también requerimientos funcionales. Nosotros sólo modelaremos tareas unitarias que impliquen la interacción de un actor con el sistema.

El diagrama de casos de uso se construye a partir de alguna representación visual elaborada durante la captura de requerimientos del negocio:

- Diagrama de contexto:
  - Cuando el punto de partida es este diagrama, la identificación de actores es trivial.
  - El mayor desafío es descubrir tareas a partir de los flujos de datos. La pregunta a responder para descubrirlos es: ¿Qué necesidades debe satisfacer cada una de las entidades externas con la ayuda del sistema?
  - Si el diagrama es demasiado complejo puede dividirse en varios diagramas parciales. En conjunto modelará al sistema completo.
  - Al validar el diagrama se identificarán tareas ausentes u oportunidades para refinar el modelo.
- Árbol o lista de características:
  - Cuando el punto de partida es este, es preferible comenzar por las características y asociarlas a tareas que cada uno de los usuarios pueden realizar en una sesión de trabajo.
  - En problemas muy simples si el árbol es detallado, es probable que todos los casos de uso correspondan a características.
  - De un árbol pueden salir varios diagramas, que en su conjunto modelarán al sistema completo.

Si no se elaboraron representaciones gráficas, el trabajo es un poco más complejo para construir el DCU. Las alternativas son:

- el análisis de documentos para identificar actores, tareas y sus asociaciones: es un buen punto de partida para comenzar si no hay representaciones visuales. Dependiendo del tipo de documento, se comenzará identificando actores o tareas.
  - Si se utiliza el organigrama, por ejemplo, el analista debería hacerse las siguientes preguntas:
    - ¿Quién notifica al sistema cuando se produce un evento en el entorno?
    - ¿Quién recibe información del sistema?
    - ¿Quién brinda información al sistema?

- La lista de partes interesadas puede servir para identificar roles de usuarios, pero no debemos olvidar que un actor puede ser también otro sistema.
- la descripción de los pasos de un proceso, que permiten también identificar tareas y actores
  - ¿Qué cambios del entorno va a detectar el sistema? ¿Con qué frecuencia?
  - ¿Qué información recibe desde el exterior?
  - ¿Qué información generada por el sistema se comunica al entorno?
  - ¿Qué actores interactúan con el sistema?
  - ¿Qué tareas realizan cada uno de los actores a partir de la interacción con el sistema?
- partir de un conjunto de escenarios específicos: permiten identificar actores y casos de uso a partir de instancias específicas de caso de uso.
  - La observación es una técnica de captura que suele estar acompañada por la especificación de escenarios. A partir de instancias específicas de las interacciones con el sistema, el analista identifica actores y generaliza casos de uso. Durante la generalización se reúnen varias instancias observadas en un mismo caso de uso.
  - Un caso de uso modela a un conjunto de instancias de escenarios.
  - Es posible reconocer escenarios actuales, futuros, de validación y de entrenamiento.
  -
- un conjunto de reglas disparadores de acciones (para el caso de desarrollo de sistemas de tiempo real o embebidos, en los cuáles las representaciones gráficas no son fáciles de realizar, o no son totalmente útiles o aptas para su representación).

Al elaborar un diagrama de casos de uso, es posible combinar 2 estrategias:

- bottom up: Enfoque centrado en el usuario. Los usuarios enumeran las tareas que necesitan ejecutar con el sistema. Cada una de estas tareas se transforma en un candidato a caso de uso.
- top down: Enfoque centrado en el producto. Se identifican los procesos que el sistema debe soportar, y luego se define una lista de tareas para cada proceso. Cada tarea que implique una interacción, va a transformarse en un caso de uso.

La combinación de las 2 estrategias evitará pasar por alto tareas o interacciones.

## Historias de usuario

En las metodologías ágiles el usuario tiene un rol protagónico. La captura y la especificación se integran para la elaboración de historias de usuario.

Una historia de usuario es una narración breve, realizada normalmente por un usuario directo o indirecto que describe una característica del nuevo sistema desde su perspectiva.

La característica es una funcionalidad valiosa, o puede derivar en un conjunto de funcionalidades valiosas para el usuario.

Está escrita desde su perspectiva, y debe incluir su rol, la necesidad y su propósito.

## Prototipos

Son simulaciones en papel o en software, de una parte del sistema. El analista busca simular los usos que el usuario va a hacer del producto.

Muestra la navegabilidad de la interfaz de usuario de forma estática o dinámica.

Esta técnica se emplea cuando el usuario no tiene muy claro lo que necesita o no puede expresarlo claramente. Sirve para que el usuario exprese sus necesidades e identifique obstáculos.

Durante el desarrollo de requerimientos de un sistema complejo e innovador el usuario con frecuencia necesita algo tangible para explorar. Puede expresar sus problemas, algunas necesidades, pero el analista identifica imprecisiones y omisiones que impiden avanzar.

Un mecanismo que permite evitar que se inviertan más recursos que los necesarios para el desarrollo de un producto es el prototipado. En ingeniería aeronáutica un prototipo puede ser un dibujo o una maqueta, pero también un avión real, que vuela para probar un nuevo modelo, antes de que se produzca en serie.

En ingeniería de software un prototipo por lo general es un experimento, una simulación, probablemente parcial, del producto final. Se utilizan particularmente para capturar, documentar y validar requerimientos para la interfaz de usuario.

Elaborar un prototipo tiene un costo. Probablemente solo una parte del producto requiera de un modelo concreto y es importante tener claro para qué y por qué construimos un prototipo, qué vamos a prototipar, y cómo vamos a construirlo.

### *¿Para qué y por qué?*

Para el desarrollo de requerimientos, porque permiten alcanzar reducir la incertidumbre, establecer acuerdos, identificar malentendidos e imprecisiones y ambigüedades, encontrar errores y omisiones, evaluar la calidad y validar los requerimientos a un bajo costo.

Para el diseño, porque permiten explorar distintas alternativas, vislumbrar el producto final, optimizar la usabilidad, evaluar técnicas potenciales y confirmar la visión del producto antes de construir la solución final.

Para el diseño, porque permiten explorar distintas alternativas, vislumbrar el producto final, optimizar la usabilidad, evaluar técnicas potenciales y confirmar la visión del producto antes de construir la solución final.

### *¿Qué?*

Es fundamental que todos los participantes tengan claro si es una herramienta exploratoria y se va a descartar o va a utilizarse como parte del producto a entregar. Si el prototipo se entrega después de un plazo corto el cliente va a estar tentado de agregar algunas funcionalidades y transformarlo en una primera versión del producto final. Si se implementa como una herramienta descartable hay que disuadirlo.

El analista debe decidir acerca del alcance del prototipo. Las alternativas son una maqueta o una prueba de conceptos.

Una maqueta se enfoca en capturar y documentar la interfaz del usuario, las pantallas y la navegación, probablemente implemente poco o incluso ninguna funcionalidad. El usuario puede juzgar si podrá hacer su trabajo con el sistema y refinar sus requerimientos, no muestra el trabajo que hará el sistema. Cuando elige un control probablemente no provoque ningún efecto, o el efecto será un mensaje.

Por ejemplo, completar un formulario para generar un reporte. Algunos atributos tienen valores por omisión, en otros se despliegan listas de opciones. Cuando el usuario elige aceptar no se genera el reporte o se genera una simulación con datos ficticios porque probablemente el modelo de datos aún no se haya diseñado. Evidentemente si la

funcionalidad no está implementada pueden pasar desapercibidas reglas que el prototipo no implementa porque están ligados a requerimientos funcionales.

Alternativamente el alcance puede implementar algunas funcionalidades para explorar y validar si la especificación de un conjunto de requerimientos es correcta completa y consistente y orientar el trabajo de diseñadores e implementadores. Implementa parte de la funcionalidad atravesando todas las capas técnicas, desde la interfaz hasta la construcción, funciona como va a funcionar el sistema real. Se crea un ambiente simulado que permite evaluar un esquema de base de datos, confirmar la solidez de una solución en la nueva o probar requisitos críticos de sincronización.

El analista debe decidir acerca del uso futuro del prototipo. Las alternativas son modelo descartable o evolutivo. El primero se tira luego de ser usado para desarrollar requerimientos o se incluye en la documentación, pero no se usa como punto de partida para el desarrollo del producto. Un prototipo evolutivo se transforma en el producto final a través de una secuencia de iteraciones.

Claramente relacionado con el alcance y el uso futuro, el analista decide si el formato va a ser electrónico o un boceto. Un boceto es un dibujo en papel, en una pizarra o a lo sumo con alguna herramienta de dibujo. Un prototipo electrónico es un producto de software que modela algún aspecto de la solución.

Cada prototipo tendrá una combinación específica de estos atributos. Por ejemplo, es posible idear una maqueta de papel descartable con dibujos sencillos de posibles pantallas. O construir una prueba de concepto electrónica evolutiva, de modo que cada iteración genera un producto entregable. Por supuesto ciertas combinaciones no tienen sentido, por ejemplo, una prueba de conceptos evolutiva en papel.

Un prototipo descartable podría estar enfocado en una porción del sistema sobre la que hay un alto nivel de incertidumbre, ambigüedad o un alto riesgo de que se hayan omitido requerimientos. Si está enfocado solo en los requerimientos funcionales no debería evolucionar para transformarse en el producto porque probablemente no se consideraron factores de calidad y restricciones.

### **Construcción de un prototipo**

En numerosas aplicaciones el prototipo puede elaborarse a partir de la interfaz de un producto similar. Consideremos por ejemplo una aplicación de comercio electrónico para una empresa de electrodomésticos. Hay miles de aplicaciones de ese tipo, el punto de partida podría ser elegir una y modificar la identidad visual, manteniendo la estructura y navegabilidad. De hecho, hay herramientas desarrolladas que pueden usarse para ofrecer un catálogo en línea, un carrito de compra, seleccionar el pago, etc.

Otra posibilidad, como describimos a continuación es partir del diagrama de casos de uso. En esta alternativa, después de elaborar el diagrama de casos de uso y antes de producir el diseño detallado de la interfaz de usuario, se construye un prototipo, total o parcial.

Cuando el analista elabora el diagrama de casos de uso se concentra en las tareas del usuario. A continuación, puede elaborar un mapa de diálogo que describa la navegabilidad y un prototipo que permita validar su propuesta. Una vez validado el prototipo, se elaboran las especificaciones detalladas de casos de uso que podrán incluir requerimientos para la interfaz de usuario o remitir a documentos que especifiquen detalladamente este interfaz.

Un mapa de diálogo es un grafo en el cual cada nodo representa una posible página y los arcos modela la navegabilidad. Quizá el sitio web no termine brindando exactamente una página para cada nodo del mapa de diálogo, algunas páginas pueden condensarse y otras expandirse, pero en esta etapa el foco está en la navegabilidad.



Claramente se trata de problemas para los cuáles no hay, en ese momento, una solución ni cercanamente similar a las expectativas del cliente. El cliente necesita algo tangible para decidir si es lo que necesita o al menos para criticarlo e identificar lo que no necesita. Si la hubiera, se podría partir de ella.

El beneficio de los prototipos es que pueden reducir el gap entre la visión del producto del cliente y la interpretación de los desarrolladores. Los prototipos conducen un diálogo a través del cual los participantes pueden acordar requerimientos. Los prototipos también son valiosos para validar requerimientos.

El riesgo es construir todo un espacio de soluciones hasta hallar la que satisface al cliente. Otro riesgo es lograr identificar los requerimientos funcionales, pero no los factores de calidad y restricciones. Un prototipo descartable enfocado solo en los requerimientos funcionales no debería evolucionar para transformarse en el producto.

### **Evaluar un prototipo**

La evaluación del prototipo de una interfaz está directamente relacionada con el testing de usabilidad.

Observe al usuario interactuar con el prototipo y evalúe la usabilidad.

Elija usuarios representativos por su nivel de experiencia, por su perspectiva o por su rol.

Elabore una encuesta simple que el usuario pueda completar inmediatamente después de usar el prototipo.

Más que preguntarle al usuario qué espera de la interfaz, el analista intenta descubrir su comportamiento instintivo. Si encuentra rápidamente lo que busca o tiene que recurrir a la ayuda. Los mensajes son particularmente importantes, si el usuario completa un formulario lo envía y no recibe respuesta probablemente quede desconcertado preguntándose si la información quedó registrada.

Claramente la experiencia con la interfaz con otros sistemas puede ser facilitar el proceso o bien obstaculizarlo, en particular hay un cambio drástico de tecnología. Pensemos en la adaptación de los adultos mayores a los teléfonos sin teclado. El prototipo tiene que ser evaluado entonces desde diferentes perspectivas por representantes de diferentes clases de usuarios con diferentes niveles de experiencia.

### **Riesgos**

- **Esfuerzo excesivo:** La construcción de prototipos demanda recursos, la idea es reducir el tiempo invertido, si el experimento demanda más recursos que el producto en sí mismo no tiene sentido. Esto ocurre cuando se intenta simular el sistema completo
- **El administrador del proyecto no incluye el prototipo en el plan:** Es fundamental que el administrador del proyecto estime los recursos que se invertirán en el desarrollo y evaluación de prototipos y que el equipo de desarrollo se ajuste al plan. No buscar la perfección en los prototipos descartables, si sirven para validar requerimientos es suficiente, no tienen que ser eficientes, confiables, seguros.
- **Falsas expectativas:** Si el usuario recibe un prototipo unos pocos días después de las primeras entrevistas, tenderá a pensar que en un par de semanas más va a tener el sistema funcionando. Tiene que quedar claro que no es así. El equipo de desarrollo debe resistir la presión y explicar que el prototipo es un experimento y no es posible ponerlo en producción tal como está. Convertir un prototipo que se pensó como descartable en el producto final difícilmente tenga un final feliz. La presión es

particularmente fuerte cuando hay un producto de la competencia ganando mercado rápidamente y esperar puede implicar perder justamente un segmento del mercado. Los participantes deben acordar claramente el alcance, si es un prototipo descartable o evolucionará hasta transformarse en el producto final. La evaluación de un prototipo en un ambiente de desarrollo, seguramente no considere factores de calidad como rendimiento, disponibilidad, robustez.

- Distracción con detalles irrelevantes: Cuando el prototipo aspira explorar funcionalidad debería destinarse poco esfuerzo a la interfaz, se reduce el riesgo de que el usuario aumente sus expectativas respecto al tiempo de desarrollo, pero como contrapartida el riesgo puede ser que se detenga en detalles irrelevantes como el tipo de letra, el contraste de los colores. También es un factor de distracción que los datos no tengan sentido, si el sistema va a gestionar datos de un hospital de niños la edad de un paciente no debería ser 65, probablemente no afecte al prototipo, pero distrae al usuario y se pierde el foco.
- El prototipo reemplaza a la especificación de requerimientos Cuando los participantes exploran los requerimientos a través de un prototipo o una secuencia de prototipos y llegan a un acuerdo, aún deben completar la documentación los requerimientos a través de diagramas y descripciones textuales. El prototipo puede formar parte de la especificación, pero no es la especificación.

## Requerimientos del producto

### Requerimientos funcionales

Los requerimientos funcionales modelan el comportamiento del sistema, esto es, las funciones que los desarrolladores deben implementar para que el usuario pueda realizar sus tareas. Describen el comportamiento que el sistema debe tener ante cada condición específica.

Se expresan a través de la entrada que recibe el sistema, la operación que realiza, y la salida esperada en cada uno de los escenarios posibles.

El analista captura directamente o deriva a partir de los requerimientos del usuario, la funcionalidad que el sistema debe brindar para satisfacer al usuario.

Cuando el foco del desarrollo de requerimientos está centrado en el producto el analista captura y especifica funcionalidades en base a las características establecidas a partir de una oportunidad del mercado o idea de producto.

En un enfoque centrado en el usuario el analista deriva los requerimientos funcionales a partir de los requerimientos del usuario, los atributos de calidad que le resultan valiosos y las reglas del negocio. La especificación combina diferentes notaciones, textos y representaciones visuales con distintos niveles de abstracción.

En ambos enfoques el analista puede construir el diagrama de casos de uso, elaborar especificaciones breves y detalladas que establecen requerimientos funcionales y completar el modelo de análisis con representaciones visuales. Algunas técnicas de representación visual que permiten modelar el comportamiento del sistema son:

### **Diagrama de actividad**

Un diagrama de actividad es una técnica adecuada para representar visualmente el flujo lógico de un proceso desde un punto inicial hasta un punto final. Permiten modelar acciones

que se realizan en forma concurrente, en secuencia y flujos alternativos que dependen de un evento, condición o decisión.

El diagrama de actividad muestra los puntos de decisión y las condiciones que quiebran el flujo normal y el flujo alternativo. El estado final va a ser el mismo, el diagrama modela dos escenarios que permiten completar la tarea del actor, pero las actividades para lograrlo son diferentes. Es útil para validar requerimientos porque permiten que el usuario visualice claramente el flujo principal, los flujos alternativos y paralelos.

Un diagrama de actividad también permite modelar el orden entre diferentes casos de uso. Cada actividad representa en esta alternativa a una tarea dentro de un proceso. En general, utilizaremos diagramas de actividad para modelar las subtarefas realizadas dentro de un único caso de uso, o para modelar el orden entre distintas tareas o casos de uso ejecutadas por un mismo actor.

La funcionalidad derivada de un caso de uso va a quedar especificada por una descripción breve, una descripción detallada o un diagrama de actividad. Mientras que en una descripción vamos a aludir a reglas del negocio, protocolos de comunicación o interfaces gráficas a través de rótulos que referencian a otras especificaciones, en los diagramas nombraremos a las tareas o subtarefas de manera más sucinta.

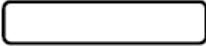
Elegimos una representación visual cuando buscamos enfatizar el flujo de control, incluso modelar paralelismo. Usamos una descripción cuando nos interesa explicitar la vinculación del caso de uso con otras especificaciones.

Componentes	
Actividad	
Flujo secuencia	
Decisión/Unión	
Concurrencia	
Inicio	
Terminación	

### Diagrama de carriles

Un diagrama de carriles (swimlane) representa las actividades que conforman un proceso, separándolas en carriles. Cada carril corresponde a un actor. El proceso puede ser un flujo de trabajo o de negocios o las interacciones entre el sistema y los actores. Algunas de las acciones pueden realizarse fuera del software.

Esta técnica permite especificar el flujo de un conjunto de casos de uso o incluso un diagrama de casos de uso completo. La construcción del diagrama comienza identificando a los actores que intervienen en el proceso que se va a modelar y asociando cada uno de ellos a un carril.

Componentes	
Carriles para actores	
Actividad	
Flujo secuencia	
Decisión/Unión	
Inicio	
Terminación	

### Tabla de eventos y respuestas

En los sistemas en tipo real y sistemas embebidos suele ser difícil reconocer requerimientos de usuarios, de modo que el modelo de casos de uso, los diagramas de carriles o de actividad no son adecuados para este tipo de aplicaciones.

Una alternativa es identificar los eventos externos que el sistema debe ser capaz de percibir y ante los cuales debe reaccionar. Un evento es toda acción que se realiza en el entorno y que provoca alguna respuesta del sistema.

Una tabla de eventos-respuesta especifica una lista de eventos y el comportamiento que el sistema debe exhibir en respuesta a cada uno de ellos. Los eventos pueden ser:

- De la aplicación: son eventos que se disparan como consecuencia de acciones realizadas por el usuario que provocan un diálogo con el sistema
- Señales: son eventos que generan otros dispositivos de hardware o de software y que el sistema percibe.
- Temporales: son eventos programados para dispararse en ciertos horarios o después de transcurrido un intervalo de tiempo respecto a una acción.

El diagrama de contexto es útil para detectar los actores que pueden disparar eventos.

El diagrama de estados es útil para identificar los distintos estados en los que se puede encontrar el sistema al percibir un evento. A partir de esta información el analista identifica cómo debe reaccionar el sistema ante cada combinación estado-evento, incluyendo excepciones.

La tabla va a complementarse con especificaciones en lenguaje natural lo suficientemente expresivas como para que el desarrollador pueda interpretar con precisión los requerimientos funcionales e implementarlos y el responsable del testing pueda verificar el sistema.

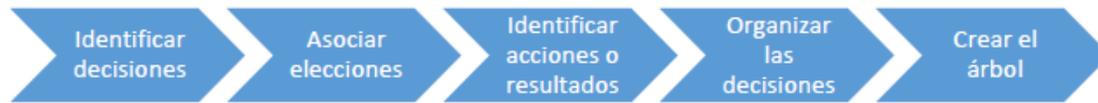
### Árbol y tabla de decisión

En muchas ocasiones el comportamiento de un sistema de software no queda determinado por las interacciones con los actores, sino que depende de complejas combinaciones de condiciones. Los requerimientos funcionales deben especificar lo que el sistema debe hacer en cada combinación posible.

El árbol de decisión es una técnica adecuada para modelar la lógica de las decisiones.

Los protocolos de actuación pueden modelarse con esta técnica, por ejemplo, en un servicio centralizado de atención de llamadas telefónicas. Los requisitos que determinan el resultado de un procedimiento de admisión o de evaluación de una solicitud, como por ejemplo un crédito, pueden modelarse también con un árbol de decisión.

La creación del árbol requiere:



Al organizar las decisiones es importante considerar las más restrictivas en los niveles superiores. La primera versión del árbol es probable que incluya la misma elección dos o más veces, es posible entonces reorganizar las ramas arrastrando las decisiones asociadas a estas elecciones a los niveles más altos del árbol. Cada decisión estará asociada a dos o más elecciones.

Una tabla de decisión es también una técnica adecuada para modelar múltiples condiciones porque permite verificar que todas las combinaciones han sido consideradas. La tabla lista todos los valores o elecciones posibles que pueden tomar cada una de las decisiones que pueden influir en el comportamiento e indica las acciones que el sistema debe tomar en respuesta de cada combinación de decisiones. Las decisiones pueden ser preguntas o frases ligadas a valores binarios (si-no o falso-verdadero) o con múltiples elecciones.

El árbol de decisión es más adecuado cuando se pretende especificar el orden de las decisiones. Una alternativa es construir primero la tabla para representar todas las combinaciones posibles, descartar las que no son relevantes y luego construir el árbol organizando las ramas adecuadamente.

Ambas técnicas de representación son particularmente útiles para representar reglas del negocio, en particular disparadores de acción. Deben combinarse con otras ya que no resultan adecuadas para modelar el flujo en un proceso.

Si el producto va a reemplazar a otro el analista puede comenzar dibujando el árbol o la tabla con las combinaciones que contempla el sistema actual y utilizarlo para descubrir las que faltan.

El lenguaje natural no es una alternativa apropiada para modelar estas aplicaciones. Es muy fácil pasar por alto combinaciones y como consecuencia surgen requerimientos ausentes y ambigüedades.

### **Diagrama de flujo de datos**

El diagrama de flujo de datos (DFD) modela el comportamiento de un sistema de información en base a los datos que recibe como entrada y la información que produce como resultado del procesamiento. El analista debe entender cómo deben procesarse los datos para generar la salida, pero antes de eso es esencial que comprenda cómo van a ser utilizados estos resultados. El diagrama de contexto es un diagrama de flujo de datos de alto nivel.

Esta técnica de representación visual es particularmente adecuada para modelar reglas del negocio, en particular cómputo. Deben combinarse con otras ya que no modelan decisiones y muchos analistas consideran que tampoco debería especificarse el orden de los procesos. La creación de un diagrama de flujo de datos implica:



Tratándose de un modelo que no solo se enfoca en los procesos sino también en los datos, también es posible identificar primero los datos y luego los procesos. El mapa de ecosistema puede ser una herramienta útil como punto de partida para identificar datos, si se construyó previamente. En cualquier caso, es importante elegir cuidadosamente los nombres de cada una de las componentes para que la validación con el usuario sea efectiva.

Los procesos modelan la transformación de los datos de entrada en los datos de salida. El nombre del proceso describe la función que realiza: “verbo en infinitivo + objeto”. Los procesos se numeran de acuerdo al nivel al que pertenecen. Algunos autores recomiendan numerar los procesos, otros consideran que no se debe establecer el orden de los procesos, el foco está en las transformaciones.

Los flujos de datos modelan los canales a través de los cuales fluyen los datos. Con frecuencia se representan no con una flecha sino con un arco con la dirección que entra o sale de un proceso. Conectan a cada proceso con:

- un terminador
- un almacenamiento
- otro proceso

Los flujos de datos que conectan un proceso con un terminador o con otro proceso pueden tener un nombre.

Cada almacenamiento representa un conjunto de datos que perdura almacenado. Los datos se almacenan por un requerimiento de usuario o para obtener datos que son requeridos por los usuarios. Tengamos en cuenta que el diagrama de flujo de datos es también un modelo de datos, de modo que la estructura de los almacenamientos se describe en el Diccionario de Datos.

Los almacenes de datos son conceptuales y podrían no representar almacenes de datos físicos reales en la solución. Además, dos entidades diferentes podrían mantenerse en almacenamientos físicos diferentes. Cuando es necesario referenciar atributos específicos de una entidad puede utilizarse la notación con punto <objeto.atributo> o directamente el nombre del atributo o conjunto de atributos.

Un proceso puede leer o escribir en el almacenamiento. Un proceso es una entidad activa, un almacenamiento es una entidad pasiva.

Los terminadores representan entidades que producen o consumen datos en el contexto externo al sistema. Pueden ser personas, organizaciones o sistemas. Los flujos entre los terminadores y los procesos modelan la interface del sistema. Las relaciones entre terminadores no se muestran en el DFD.

La creación puede hacerse en niveles comenzando con un nivel alto de abstracción al construir y validar el diagrama de contexto. Para modelar problemas complejos el DFD se organiza en 3 o 4 niveles:

#### **Nivel 0: Diagrama de contexto**

- Modela todas las interacciones que realiza el sistema con entidades externas del entorno.
- Incluye un solo proceso que representa al sistema.
- Incluye a todas las entidades externas que son fuente y destino de los datos.

- Todos los flujos de datos denotan relaciones entre el sistema y las entidades externas.
- No incluye almacenamiento

**Nivel 1: División del sistema en subsistemas**

- Modela todos los procesos que conforman el proceso principal.
- Los procesos se relacionan a través de unidades de almacenamiento o entidades externas, considerando que la mayor parte de los datos se almacenan.

**Nivel 2: División en módulos**

- Modelan excepciones a los caminos principales
- Aumenta progresivamente el nivel de detalle incorporando aspectos de la solución.
- Aparecen flujos entre procesos.

**Nivel 3: Detalle o expansión de los módulos**

- Modelan excepciones a los caminos principales
- Aumenta progresivamente el nivel de detalle incorporando aspectos de la solución.
- Aparecen flujos entre procesos.

Cada vez que se avanza de nivel se analiza el balance del modelo:

Todos los flujos que de entrada al proceso de nivel n deben aparecer como flujo de entrada de algún proceso de nivel n+1

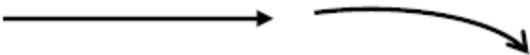
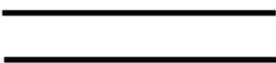
Todos los flujos que de salida del proceso de nivel n deben ser flujo de salida de algún proceso de nivel n+1.

**Conexiones permitidas:**

ENTIDAD - PROCESO  
 PROCESO - PROCESO  
 PROCESO – ALMACÉN

**Conexiones no permitidas:**

ENTIDAD - ENTIDAD  
 ALMACÉN - ALMACÉN  
 ENTIDAD - ALMACÉN

Componentes	
Entidades externas	
Flujo de datos	
Procesos	
Almacenamiento	

**Diagrama y tablas de transición de estados**

Un diagrama de estados o diagrama de transición de estados modela todos los estados que puede atravesar una entidad en respuesta a los eventos que percibe. La entidad puede ser

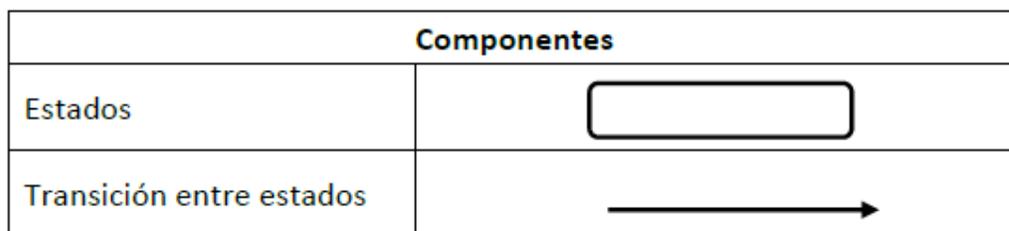
un objeto de una clase, un subsistema o el sistema completo. Un estado representa una condición que una entidad satisface en un momento específico.

Son útiles cuando una entidad participa en diferentes casos de uso y en la transición de uno a otro cambia de estado. Son concisos y permiten verificar precisión y completitud. No son adecuados para modelar la colaboración entre un conjunto de entidades dentro de un caso de uso ni el flujo de datos o acciones dentro de un proceso.

Una notación alternativa para modelar la transición de estados de una entidad a lo largo de un proceso, son las tablas de estado. Una tabla de estados es una matriz en la cual la primera fila y la primera columna identifican a todos los estados posibles. Las celdas que se forman al cruzarse estados se rotulan con los eventos, siempre que el cambio de estado sea posible.

Tanto en el diagrama como en la tabla, cada estado es único y en cada momento una entidad está en un estado específico. En general no son técnicas complementarias sino más bien alternativas, ya que ambas modelan las transiciones de un solo paso que atraviesa una entidad.

Ambas técnicas permiten representar las transiciones de una entidad y por lo tanto pueden elaborarse como parte del modelo de datos. Hemos incluido esta técnica junto con otras que modelan comportamiento ya que las transiciones de un estado a otro surgen del flujo de las actividades de una tarea unitaria o de un proceso formado por varias tareas o etapas.



Alternativamente el analista puede utilizar alguna de estas técnicas para construir una representación visual y derivar luego el diagrama de casos de uso. Los diagramas de flujo de datos y los diagramas y tablas de transición de estados son también modelos de datos. Permiten justamente representar el comportamiento con relación a las transformaciones o cambios que experimentan las entidades a lo largo de un proceso.

Ninguna de las técnicas de representación visual es lo suficientemente expresiva como para modelar el comportamiento completo desde diferentes puntos de vista, pero en conjunto se complementan y permiten validar la especificación, identificar ausencias y resolver ambigüedades. No es necesario utilizar todas las técnicas en cada proyecto, el analista decidirá cuáles son más adecuadas dependiendo del tipo de producto y del conocimiento de las partes interesadas.

## Requerimientos para los datos

Los datos empiezan a reconocerse en las primeras iteraciones, cuando se establecen los requerimientos del negocio, y se hacen más evidentes cuando se capturan los requerimientos del usuario y los requerimientos funcionales.

Durante el análisis de una sesión de captura, el analista identifica los sustantivos que aparecen en las necesidades y requerimientos formulados y nombra entidades o atributos de entidades relevantes en el problema.

Un sistema de información permite recibir, almacenar, administrar, controlar, procesar y/o transmitir datos.

Los modelos de datos elaborados por el analista tienen un alto nivel de abstracción, se extienden y se completan durante la etapa de diseño del sistema, y se transforman durante la implementación en una base de datos.

Un modelo de datos especifica los datos que van a ser almacenados por el sistema en forma más o menos permanente y sus relaciones. Un modelo conceptual de datos es una representación abstracta de las entidades, no considera el comportamiento ni a las restricciones de implementación.

En muchas aplicaciones el modelo de datos tiene un rol protagónico, la mayoría de los requerimientos funcionales se refieren a cómo procesar y consultar datos. En los sistemas de información la especificación de un modelo adecuado para los datos es el aspecto central, tanto en la etapa de desarrollo de requerimientos como posteriormente durante el diseño. En estas aplicaciones gran parte de la funcionalidad está vinculada a consultas sobre datos almacenados y la generación de reportes.

En otras aplicaciones, como por ejemplo los sistemas en tiempo real, el rol protagónico lo tienen los requerimientos funcionales. Por supuesto el sistema recibe datos de entrada, produce datos de salida y puede mantener datos almacenados, pero las necesidades de los usuarios están más bien vinculadas con el comportamiento.

Un mismo problema puede modelarse de diferentes maneras y utilizando diferentes técnicas. El modelo debe ser coherente, es decir, el conjunto de representaciones que especifican el problema desde diferentes perspectivas deben ser consistentes entre sí. En particular el modelo de datos debe ser consistente con el modelo de comportamiento.

## Diagramas de clases

Un diagrama de clases muestra la estructura estática de los elementos relevantes del sistema y sus relaciones, sin considerar el tiempo. Cada clase es un patrón que modela los atributos y el comportamiento de un conjunto de objetos relevantes del problema. Aunque se llama diagrama de clases, las relaciones entre las clases son tan importantes para el modelo como las clases mismas.

Durante el desarrollo de requerimientos el analista se concentra en primera instancia en identificar los conceptos relevantes y el modo en que se relacionan. El diagrama de conceptos del negocio o diagrama de conceptos del dominio es un diagrama de clases básico, cada concepto es representado a través de una clase. Tiene que ser muy simple porque va a ser compartido con el usuario o cliente.

El diagrama de conceptos del negocio modela el problema, no la solución, puede construirse como parte de los requerimientos del negocio. Muchos de los conceptos van a definirse en primera instancia en el glosario, el diagrama evolucionará luego para incluir atributos y luego se refinará la especificación en el diccionario de datos. Recién en la etapa de diseño el diagrama de clases va a modelar el comportamiento.

En un sistema de mediana o gran escala es imposible incluir el conjunto completo de clases en un diagrama. El conjunto de diagramas de clases puede distribuirse en paquetes.

### **La relación de asociación**

La asociación es una relación estructural, estática entre clases. Las asociaciones se explicitan en el diagrama de clases estableciendo un arco entre las clases relacionadas.

Cada asociación puede tener un nombre y opcionalmente una flecha que permite navegar entre las clases, es decir, nos indica la dirección para “leer” el nombre.

La aridad de una asociación establece cuántas clases participan en la relación.

La multiplicidad de una relación indica cuántas instancias de una clase estarán vinculadas a una instancia de la clase asociada. La cantidad puede ser:

- 1
- 1 o más (1..\*)
- 0 o 1 (0..1) (opcional)
- 0 o más (0..\*) (opcional)
- Un rango específico (m..n)
- El asterisco denota a cualquier número de entidades
- 

Tanto la multiplicidad como el nombre de las relaciones son elementos opcionales, que cada equipo de desarrollo decide si incluye o no en el modelo. Nosotros adoptamos la convención de modelar la multiplicidad, pero no nombrar a las relaciones.

Cuando la relación es 1 a 1 existe una dependencia entre clases: Es probable que las instancias asociadas de ambas clases se creen simultáneamente. Una instancia de una clase no puede existir sin la instancia de su clase asociada.

:

Una clase **asociación** es una clase que modela la asociación entre otras clases y tiene atributos propios. Por lo general se presenta en las relaciones con multiplicidad muchos a muchos.

La **agregación** es una relación más fuerte que la asociación porque el comportamiento de una instancia de una clase depende del comportamiento de la instancia relacionada.

La **composición** es una relación aún más fuerte que la agregación porque la existencia de una instancia de una clase depende de la existencia de la instancia de la clase relacionada.

Las relaciones de agregación y composición permiten modelar la dependencia entre clases que forman parte de un todo. Como en esta etapa no especificamos comportamiento ni aspectos dinámicos, en el modelo de datos no vamos a considerar agregación y composición.

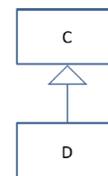
Tampoco vamos a aplicar patrones de diseño durante el desarrollo de requerimientos, justamente porque corresponden a la etapa de diseño de la solución. El modelo de datos va a ir evolucionando progresivamente, desde la representación de conceptos y sus relaciones de alto nivel, hasta un diagrama de clases que especifique el comportamiento de las instancias.

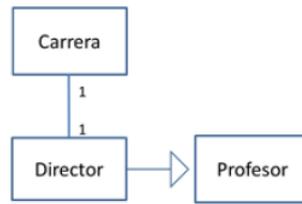
### **La relación de herencia**

La herencia es un mecanismo que establece una relación de generalización-especialización entre clases.

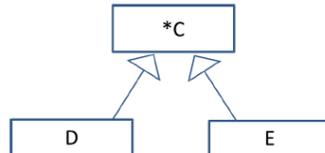
Una instancia de la clase D es también instancia de la clase C.

Una instancia de Director es una instancia especializada de Profesor





Una clase abstracta es aquella que no modela a ningún objeto del problema.



Por lo general no se especifican en la etapa de análisis porque se refieren al comportamiento y porque los objetos forman parte de los aspectos dinámicos.

### **Los atributos**

Como mencionamos antes cada clase define un conjunto de atributos y de comportamiento. Sin embargo, el comportamiento de las instancias de una clase va a especificarse en la etapa de diseño. Durante el desarrollo de requerimientos el analista identifica conceptos y el modelo evoluciona para especificar también los atributos, pero no su tipo ni visibilidad.

Un error común es aplicar el patrón de diseño state al elaborar el modelo. Reiteramos que durante el desarrollo de requerimientos elaboramos un modelo de análisis, por lo tanto, los patrones de diseño van aplicarse recién cuando la representación evolucione, justamente en la etapa de diseño. En particular el patrón state es un patrón de comportamiento y nuestro diagrama no incluye las operaciones provistas por la clase.

Observemos que los únicos valores para los atributos que estamos especificando en el modelo de análisis, son los enumerados.

El conjunto de valores va a modelarse como una clase y no como un enumerado cuando puede cambiar.

### Diccionario de datos

El diccionario de datos es una descripción textual de los datos, con una estructura basada en una plantilla. Aunque no es un modelo formal, si es importante elaborarlo aplicando una metodología rigurosa. Describe tanto los datos almacenados como los datos de entrada y salida.

El diccionario de datos mantiene un nivel de detalle que solo es posible alcanzar con lenguaje natural. Son particularmente útiles para describir casos complejos. En los diagramas es difícil especificar casos especiales o excepciones.

Es un complemento de las representaciones visuales y debe ser consistente con ellas, como así también con el Glosario. El punto de partida suele ser justamente el glosario. Mientras que la estructura del glosario es “expresión”, “definición” “sinónimos”, en el diccionario la caracterización es más completa.

Cada clase de un diagrama de clases va a estar asociada a una descripción en el diccionario de datos, aunque no es necesario completar la plantilla para cada entidad o clase. Cuando el modelo evolucione, de cada atributo se incluirá al menos el nombre y el tipo. Se puede especificar de forma opcional las reglas de validación, las fórmulas de cálculo y los valores por omisión.

Es importante resaltar que en un modelo de análisis, la especificación de una clase incluye únicamente los atributos que corresponden a tipos elementales. Cuando el diagrama evoluciona, se agregan los atributos que modelan la asociación entre clases. Estos atributos no se especifican en el diccionario de datos.

El análisis de las reglas del negocio permite identificar muchas de las reglas de validación y fórmulas. Las reglas de validación y las fórmulas pueden involucrar a varios atributos. Las reglas de validación modelan restricciones y garantizan cierto nivel de integridad en los datos.

La principal desventaja de los diccionarios de datos es que tienden a crecer desmesuradamente y son difíciles de validar. Para evitarlo no deberían describirse casos triviales. El mayor riesgo es la inconsistencia, sobre todo cuando participan diferentes analistas. En algunos proyectos el analista sólo elabora el diccionario de datos y el diseñador genera las representaciones visuales. En este caso el diccionario de datos no es un complemento de los requerimientos, sino que los requerimientos se especifican exclusivamente en el diccionario de datos.

El tiempo que invierta en crear un diccionario de datos será más que recompensado al evitar los errores que pueden surgir cuando los participantes del proyecto tienen diferentes interpretaciones o criterios acerca de los datos. Mantenerlo actualizado durante todo el ciclo de vida del producto también demanda una inversión importante, pero es fundamental no solo para recurrir a él cuando surgen diferencias en la interpretación o de criterio, sino también para reusar definiciones.

El diccionario es particularmente valioso cuando se elabora una línea de productos porque permite mantener definiciones consistentes en toda la organización. Esto reduce la complejidad de la integración y los errores entre las interfaces. De hecho, puede ser recomendable desarrollar un diccionario de datos generales de la organización y luego un diccionario específico para cada proyecto, el lugar de mantener documentos y consolidarlos cuando se producen cambios.

**Plantillas para el diccionario de datos**

Para todo flujo de entrada o salida de datos al sistema también se incluirá una descripción en el diccionario de datos. En particular, se indicará el nombre del flujo, si se trata de un flujo de entrada o de salida, y para cada componente del flujo se indicará su nombre, su tipo (puede referenciarse a una clase), y las reglas de validación (para flujos de entrada) o fórmulas de cálculo (para flujos de salida). También podría incluirse una columna para el valor por omisión, si corresponde.

Generalmente, para los flujos de entrada de datos se adopta la convención de nombrarlos con el prefijo Novedades.

Nombre Flujo:		I/O:
Nombre	Tipo	Reglas de Validación o Fórmula de cálculo

De manera similar, la plantilla utilizada para la descripción de una clase en el diccionario de datos incluye su nombre, un código que la identifica, y luego se indica cada atributo con su

tipo y sus reglas de validación. Finalmente, cuando corresponde, se indican reglas de validación o restricciones a nivel de clase.

Nombre:		Código:
Atributo	Tipo	Reglas de Validación
Restricciones sobre la clase:		

### Matriz de validación

La validación puede realizarse mediante una matriz en la que se “cruzan” casos de uso y clases y se registran diferentes tipos de operaciones. Por ejemplo “C” crear, “L” leer, “M” modificar y “B” borrar”. La generación y validación de la matriz permite detectar ausencias e inconsistencias entre casos de uso y conjuntos de instancias en el diagrama de clases.

Puede examinar varios tipos de correlaciones, incluidas las siguientes:

- Entidades de datos y eventos del sistema
- Entidades de datos y tareas de usuario o casos de uso
- Clases de objetos y casos de uso

### Requerimientos para consultas y reportes

La mayor parte de las aplicaciones generan reportes. Pueden tener un formato flexible o rígido, pueden incluir texto y gráficos. El propósito y contenido se define durante el desarrollo de requerimientos e impacta sobre la derivación de requerimientos de datos. El diagramado puede establecerlo el usuario, el diseñador o puede estar impuesto por el contexto de la organización.

Una parte del cuestionario de captura puede estandarizarse:

- ¿Qué reportes se generan actualmente?
- ¿Cuáles se agregan?
- ¿Cuáles deben ser modificados?
- ¿Quiénes son sus usuarios? ¿Para qué los usan? ¿Qué decisiones se toman a partir de la información de cada reporte?
- ¿Con qué frecuencia se usa cada uno? ¿Depende de un evento?
- ¿Cuántas hojas puede demandar como máximo?
- ¿Cuál es el soporte? Se visualiza por pantalla, se imprime, se envía por mail, se exporta a una planilla de cálculo o a algún otro formato.
- ¿Hay restricciones de seguridad o privacidad que limiten el acceso a individuos específicos o clases de usuarios?
- ¿Qué fuentes de datos tiene el reporte?
- ¿Qué parámetros puede establecer el usuario?
- ¿Qué procesamiento se requiere? ¿Subtotales? ¿Contadores? ¿Ordenamiento?
- ¿Cuál es el contenido del reporte si no hay datos para mostrar?
- ¿Puede usarse una misma plantilla para distintos reportes?
- 

El analista debería:

- Verificar que el sistema mantenga la información requerida por cada reporte o pueda computarla a partir de los datos almacenados.
- Anticipar el crecimiento de los reportes y sobre todo de las respuestas a las consultas a través de la pantalla de un dispositivo.
- Validar el Glosario considerando los términos que aparecen en los reportes.
- Identificar patrones.
- Considerar si los datos tienen que modificarse dinámicamente
- Decidir si se generan prototipos

Para poder generar un reporte que muestre, por ejemplo, en qué sesión un alumno se inscribió en una comisión de cursado, es necesario vincular las sesiones a las tareas del usuario.

Un formato que es adecuado inicialmente puede no serlo cuando el volumen de información aumenta.

En la documentación del sistema deberíamos utilizar solo términos para los cuales haya una definición precisa y preferentemente establecida formalmente. En muchas organizaciones el desarrollo de un nuevo sistema o un sistema de reemplazo son una oportunidad para definir términos de uso frecuente y no tan frecuente.

Varios usuarios pueden requerir reportes similares, definir formato común y parametrizar los aspectos que son diferentes. Por ejemplo, los distintos certificados académicos que solicitan diferentes actores.

Los reportes estáticos muestran la información que corresponde al instante en que se generó el reporte. Los reportes dinámicos se actualizan automáticamente a medida que la información cambia.

Los prototipos de reportes permiten que el usuario visualice el diagramado y decida qué características le agradan y cuáles no.

## Plantilla para reportes

<b>Código</b>	
<b>Título</b>	Nombre, posición y datos fijos y variables del título
<b>Propósito</b>	Breve descripción de la necesidad que satisface el reporte
<b>Usuarios</b>	Clases de Usuarios interesados en el reporte
<b>Privilegios</b>	Privilegios requeridos para generar cada reporte
<b>Decisiones</b>	Decisiones que cada clase de usuario puede tomar a partir del reporte
<b>Prioridad</b>	Prioridad respecto a otros reportes
<b>Latencia</b>	Tiempo de entrega del reporte al usuario que lo requiere Actualidad de los datos incluidos en el reporte
<b>Diagramado</b>	Tamaño de papel, orientación, márgenes
<b>Gráficos e Imágenes</b>	Tipos de gráficos e imágenes, tamaño
<b>Encabezamiento y pie</b>	Texto fijo y variable, Fecha Número de página, nombre del usuario que lo solicita, nivel de confidencialidad
<b>Cuerpo</b>	Criterio de selección de dato, atributos Criterio de ordenamiento, fórmulas Criterio de paginado

## Tableros de control

Un tablero de control es un reporte que aparece en pantalla o impreso y permite visualizar información relevante y consolidada referida a la organización o un proceso. Por lo general incluye algunos valores numéricos, probablemente organizados como una tabla, gráficos u otro tipo de imágenes que muestran algunos indicadores claves. Puede ofrecer también un texto breve.

En todos los reportes, pero más aún en los tableros de control, el contenido es muy importante, pero el diagramado también tiene relevancia. Tanto el contenido como el diagramado puede parametrizarse, en cuyo caso se requiere un usuario aún más entrenado.

En un tablero dinámico la información se modifica en tiempo real a medida que el estado interno de los datos se modifica. La información debe permitir detectar situaciones que requieren tomar decisiones rápidas. El usuario puede explicitar que necesita un tablero de control o puede proponerlo el analista luego de capturar los requerimientos para los reportes.

Como en el caso de los reportes es necesario capturar restricciones, privilegios, prioridades y fundamentalmente analizar si con los datos almacenados es posible ofrecer los

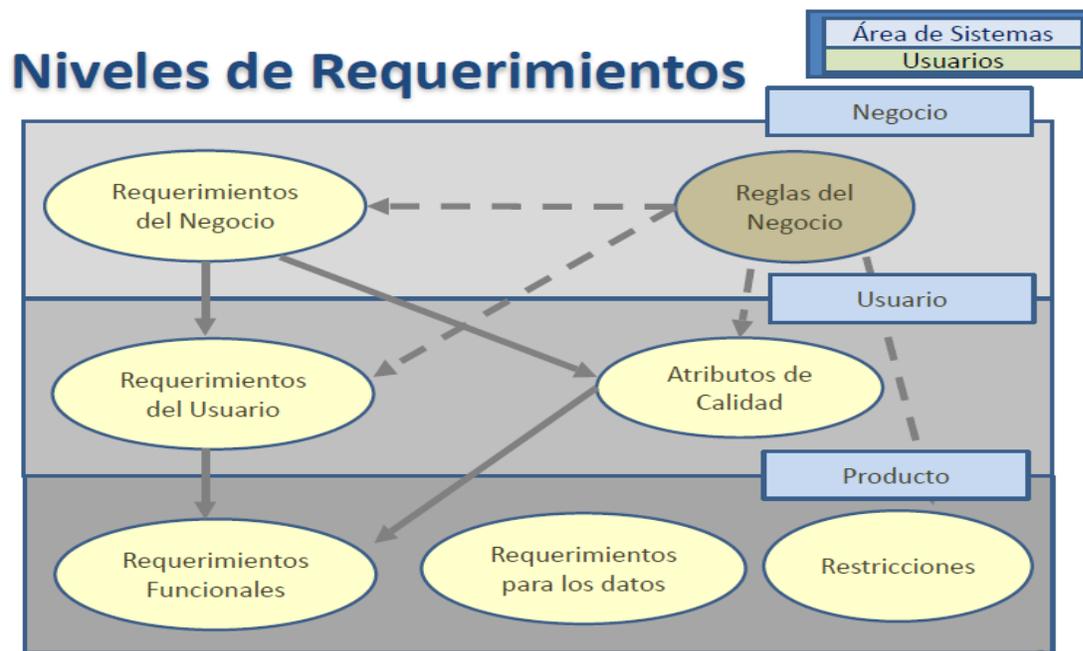
indicadores incluidos en el tablero. Los prototipos son un excelente recurso para asegurar que el estilo y diagramado satisface las necesidades del usuario.

En proyectos en gran escala o en empresas dedicadas al desarrollo de sistemas se suele asignar un especialista para el diseño de interfaces de usuario. En este caso será el responsable de diseñar los prototipos de todos los reportes, en particular el tablero de control, pero debe trabajar con usuarios representativos, con el analista y con los implementadores.

## Caso 1

Una organización demanda el desarrollo de un producto para su propio uso cuando identifica una necesidad u oportunidad de negocio que sirve como punto de partida para que el analista capture y especifique los requerimientos del negocio. Participan directivos, gerentes y patrocinadores del producto.

El desarrollo de requerimientos es un proceso iterativo incremental en el cual, en las primeras iteraciones, nos concentramos en los requerimientos de alto nivel (negocio), en el segundo nivel se pone el foco en el usuario, y en el tercer nivel se pone el foco en el producto que los desarrolladores deberán implementar.



### Nivel 1: Requerimientos del negocio

En este nivel, el analista debe:

- Capturar las reglas del negocio, de las cuáles se pueden derivar:
  - Requerimientos del negocio
  - Atributos de calidad
  - Requerimientos funcionales
  - Restricciones
- Especificar los requerimientos del negocio, de los cuáles pueden derivar:
  - Requerimientos de usuario

- Atributos de calidad

## Nivel 2: Requerimientos del usuario

Una vez que el analista acuerda con los directivos los requerimientos del negocio, si se adopta un enfoque centrado en el usuario, el analista busca identificar las distintas clases de usuarios y sus tareas. A partir de estas tareas, especifica:

- Requerimientos del usuario, de los cuáles pueden derivar:
  - Requerimientos funcionales
- Atributos de calidad: las cualidades que son valiosas para cada clase de usuario, de las cuáles pueden derivar:
  - Requerimientos funcionales

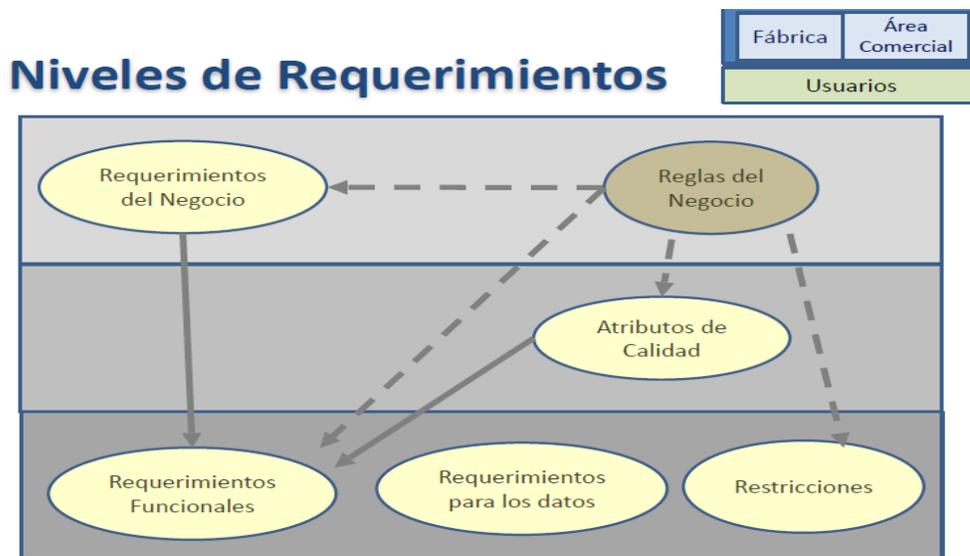
## Nivel 3: Requerimientos del producto

Antes de comenzar con el diseño, es posible empezar a pensar en la solución, definiendo:

- Requerimientos funcionales
- Requerimientos para los datos
- Restricciones que el sistema debe satisfacer

## Caso 2

Una empresa que desarrolla software para comercializar, el proceso también comienza con la definición de los requerimientos del negocio, pero en este caso es más probable que el analista adopte un enfoque centrado en el producto



## Nivel 1: Requerimientos del negocio

En este nivel, el analista debe:

- Capturar las reglas de negocio, de las cuáles se pueden derivar:
  - Requerimientos del negocio
  - Requerimientos funcionales
  - Atributos de calidad
  - Restricciones

- Especificar Requerimientos del negocio, de los cuáles derivarán directamente:
  - Requerimientos funcionales

### Nivel 2: Requerimientos del usuario

En este nivel, el analista especificará:

- Atributos de calidad, que serán valiosos en el segmento del mercado en el que se va a comercializar el producto, y los cuáles derivarán en
  - Requerimientos funcionales

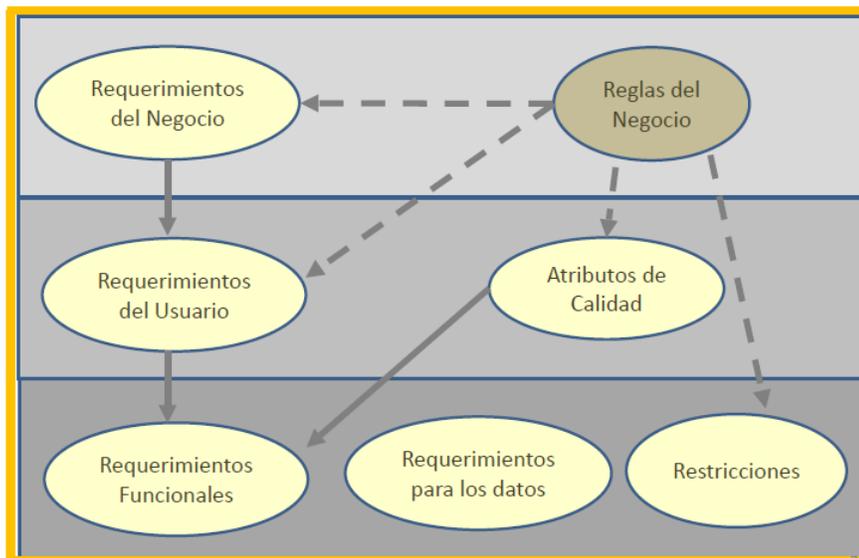
### Nivel 3: Requerimientos del producto

En este nivel, el analista especificará:

- Requerimientos funcionales
- Requerimientos para los datos
- Restricciones

## General

Cualquiera sea el ámbito de desarrollo y cualquiera sea el enfoque que se adopte, consideraremos que el desarrollo de requerimientos es un proceso iterativo incremental que produce diferentes niveles de requerimientos, y que los requerimientos del sistema incluyen a todos los niveles especificados.



## Administración de requerimientos

La administración de requerimientos incluye todas las actividades destinadas a mantener la integridad, consistencia, precisión y vigencia de los requerimientos a través del proyecto. El analista lidera la administración de requerimientos, pero no es el único responsable. Involucra el seguimiento de los requerimientos y su evolución, acompañando los cambios en las necesidades o restricciones.

No es un proceso, pero involucra las siguientes actividades:

## Gestión de cambios

### El mundo ideal

- Todos los requerimientos están especificados en documentos antes de que comience el diseño del sistema.
- Los requerimientos no se modifican hasta que el proceso de desarrollo se completa.

### El mundo real

- Durante el proceso de desarrollo de un producto pueden generarse oportunidades en el mercado , pueden producirse cambios en las reglamentaciones o pueden surgir nuevas necesidades dentro del negocio.
- Es posible que se detecten omisiones o incluso inconsistencias o ambigüedades que pasaron desapercibidas durante el desarrollo de requerimientos.

### El alcance inalcanzable

- Si los requerimientos cambian y los tiempos se alargan y aumenta la probabilidad de que surjan otros cambios.
- Si se aceptan cambios sin ajustar la planificación, que muy probablemente no pueda cumplirse.
- En ambos casos la entrega final puede ir demorándose incluso indefinidamente.

### Lo que no debería ocurrir

- El cliente propone cambios y las modificaciones se atienden sin considerar los requerimientos de los usuarios directos.
- El usuario propone cambios y se atienden sin considerar su alineación con los requerimientos del negocio establecidos por el cliente.
- El desarrollador recibe propuestas de cambio y las atiende sin la participación del analista.
- Los cambios se rechazan o se postergan para otra iteración, sin evaluar los riesgos y consecuencias de no modificar los requerimientos.
- Los cambios se aceptan sin evaluar el impacto Los cambios se aceptan y no se modifican el Plan del Proyecto de acuerdo al impacto
- Los cambios se aceptan y no se modifican los documentos de especificación de requerimientos

### El costo

Aceptar un cambio cuando el producto está en desarrollo tiene un costo . Rechazar un cambio también tiene un costo.

Todos los participantes deben entender que los cambios van a ocurrir, pero es necesario estar preparados para gestionarlos.

La gestión del cambio en sí misma, ya sea que se acepte o se rechace, también demanda tiempo y esfuerzo.

## Métricas para el esfuerzo

- El tiempo del analista:
  - organizar encuentros de captura y de validación
  - analizar documentos
  - evaluar cambios
  - crear prototipos
  - elaborar especificaciones
  - configurar las herramientas de software
  - mantener actualizado el estado de los requerimientos
  - comunicar los cambios a los participantes

## El acuerdo base

La definición del acuerdo base es el primer requisito para una gestión de cambios exitosa. El acuerdo base es un subconjunto de los requerimientos que han sido validados y aprobados al completarse el desarrollo de requerimientos. Pueden incluir requerimientos del negocio, del usuario, funcionales, no funcionales, de datos, escritos en lenguaje natural plano, plantillas o representaciones visuales.

## El plan del proyecto

Cuando el analista modifica el acuerdo base el administrador del proyecto debe modificar el plan del proyecto, para adaptarse a los cambios en los requerimientos. La modificación requiere una renegociación con los clientes en la que se evalúan distintas alternativas.

## Alternativas para ajustar el plan del proyecto

- Incorporar miembros al equipo de desarrollo o subcontratar a otra empresa para hacer parte del trabajo.
- Modificar el cronograma de entrega
- Postergar o eliminar los requerimientos de baja prioridad
- Sacrificar calidad para mantener el cronograma o modificarlo lo menos posible

## Política

El segundo requisito para una gestión de cambios exitosa es establecer y acordar una Política de Gestión de Cambios para la organización. Antes del acuerdo base los requerimientos pueden evolucionar sin seguir un procedimiento riguroso.

Después de definir el acuerdo base se aplica la política estrictamente.

1. Mantener al líder del proyecto informado de las decisiones de negocios relacionadas con el producto.
2. Designar un Grupo de Control de Cambios para el proyecto:
  - 2.1. Es responsable de analizar, evaluar y decidir acerca de los cambios
    - 2.1.1. Analizar la Aceptación
      - 2.1.1.1. Evaluar la aceptación del cambio o si este debe ser aceptado sin evaluación.
      - 2.1.1.2. El grupo analiza en este paso los riesgos y las consecuencias de no aceptar el cambio sobre los objetivos del negocio.

- 2.1.1.3. En algunas ocasiones la **opción de NO aceptar el cambio se descarta**. El cambio es determinante para el éxito del producto y debe hacerse dentro del alcance del proyecto. No aceptarlo es equivalente a cancelar el proyecto y el desarrollo del producto. Tal como está especificado el problema, el producto no va a ser una solución satisfactoria.
- 2.1.1.4. En otros casos es posible **decidir si el cambio se acepta**. Si no se acepta, el proyecto puede completarse y el desarrollo del producto termina aun cuando no satisfaga plenamente las expectativas. El cliente siempre tiene razón, no es una buena regla cuando se evalúan cambios.
- 2.1.2. Establecer y aplicar los Criterios para Evaluar alineación, factibilidad e impacto
  - 2.1.2.1. Análisis de alineación y factibilidad
    - 2.1.2.1.1. Si los cambios deben aceptarse o el grupo considera que están alineados y son factibles, la evaluación del GCB analiza el impacto del cambio.
      - 2.1.2.1.1.1. Solo deberían aceptarse nuevos requerimientos alineados con los objetivos del negocio.
      - 2.1.2.1.1.2. Si un cambio no es factible en las condiciones actuales, no puede aceptarse, al menos sin considerar cómo modificar las condiciones.
      - 2.1.2.1.1.3. Las principales condiciones son la capacidad del equipo de desarrollo y las restricciones de infraestructura.
  - 2.1.2.2. Análisis del impacto
    - 2.1.2.2.1. El análisis de impacto permite comprender las implicaciones del cambio propuesto
    - 2.1.2.2.2. Debe hacerse tanto si se está evaluando aceptar los cambios, como si estos deben hacerse imperiosamente.
    - 2.1.2.2.3. El resultado de este análisis permite renegociar los acuerdos
    - 2.1.2.2.4. Entidades identificadas en el análisis del impacto
      - 2.1.2.2.4.1. conflictos e inconsistencias entre los nuevos requerimientos y los anteriores
      - 2.1.2.2.4.2. los niveles de requerimientos afectados
      - 2.1.2.2.4.3. los modelos de análisis y secciones del SRS que deben ser modificados
      - 2.1.2.2.4.4. las piezas de diseño y componentes de software que deben ser creadas, modificadas o descartadas
      - 2.1.2.2.4.5. los prototipos que deberían desarrollarse
    - 2.1.2.2.5. Tareas del análisis del impacto
      - 2.1.2.2.5.1. Estimar el esfuerzo de modificar cada modelo de análisis, pieza de diseño o componente de software

- 2.1.2.2.5.2. Estimar el esfuerzo realizado hasta el momento que se pierde si se acepta el cambio
    - 2.1.2.2.5.3. Evaluar prioridades
    - 2.1.2.2.5.4. Calcular costos a partir del esfuerzo estimado y proponer cambios en la planificación
  - 2.1.2.2.6. Resultado de la evaluación
    - 2.1.2.2.6.1. No aceptar el cambio
    - 2.1.2.2.6.2. No aceptar el cambio en la iteración actual, analizar agregar una iteración al proceso de desarrollo del producto para la cual se establecerá un nuevo acuerdo.
    - 2.1.2.2.6.3. Aceptar el cambio en la iteración actual negociando las condiciones del acuerdo y el plan del proyecto.
- 2.1.3. Seleccionar y aplicar un Proceso de Decisión
  - 2.1.3.1. Todos los miembros del grupo votan acerca de aceptar el cambio y se adopta la decisión de la mayoría.
  - 2.1.3.2. Todos los miembros del grupo opinan y un subgrupo, con roles específicos, decide.
  - 2.1.3.3. Todos los miembros del grupo opinan y el líder de las decisiones decide.
  - 2.1.3.4. Criterios del proceso de decisión
    - 2.1.3.4.1. Financieros: costo de aceptar el cambio respecto a no aceptarlo
    - 2.1.3.4.2. Expectativas del usuario
    - 2.1.3.4.3. Ventajas competitivas
- 2.1.4. Aplicar el procedimiento de gestión de cambios
  - 2.1.4.1. A partir de que el cambio se acepta la política establece un procedimiento riguroso para manejar los cambios y un conjunto de herramientas específicas o generales .
  - 2.1.4.2. Algunos cambios en los requerimientos no afectan al plan del proyecto y pueden aplicarse un procedimiento “blando”
  - 2.1.4.3. Plantilla para la especificación de cambios
    - 2.1.4.3.1. La descripción del propósito
    - 2.1.4.3.2. El Alcance
    - 2.1.4.3.3. Los roles de las personas responsable de cada tarea, puede incluir a miembros del equipo de desarrolladores, gerentes, administradores de empresas, líderes de usuario
    - 2.1.4.3.4. La lista de estados para el requerimiento que cambia, puede modelarse a través de un diagrama de estados
    - 2.1.4.3.5. El criterio de entrada : condiciones que deben satisfacerse antes de que el proceso comience
    - 2.1.4.3.6. La lista de tareas, la evaluación de factibilidad, costo y tiempo y de pertinencia respecto a los requerimientos del negocio, las consecuencias de aceptar el cambio y los riesgos. Incluye los argumentos que justifican

- aprobar o rechazar el cambio y la prioridad que se le asigna.
      - 2.1.4.3.7. El criterio de salida , las condiciones que indican que el proceso se completó satisfactoriamente
      - 2.1.4.3.8. El reporte establece si el requerimiento de cambio fue Rechazado, Cancelado o Terminado e indica si todos los participantes fueron informados del estado final.
    - 2.1.5. Evaluar la Política de Gestión de Cambios a partir de la medición de las actividades asociadas a cambios
      - 2.1.5.1. Total de cambios solicitados, aceptados, rechazados y en tratamiento en el proyecto
      - 2.1.5.2. Total de cambios solicitados por una fuente de requerimientos
      - 2.1.5.3. Total de horas dedicadas a especificar, diseñar, implementar y verificar cambios en los requerimientos.
  - 2.2. Integrantes: Al menos uno de los miembros del grupo debe tener conocimiento profundo del negocio y la organización.
    - 2.2.1. Administrador del proyecto
    - 2.2.2. Administrador del producto
    - 2.2.3. Analista
    - 2.2.4. Desarrollador
    - 2.2.5. Responsable de testing
    - 2.2.6. Experto en aspectos del negocio
    - 2.2.7. Gerente de comercialización
    - 2.2.8. Responsable de infraestructura
  - 2.3. Roles
    - 2.3.1. Líder de las decisiones o subgrupo responsable de las decisiones.
    - 2.3.2. Responsable de mantener actualizado el estado de los requerimientos
    - 2.3.3. Responsable de renegociar los acuerdos cuando se modifican los planes como consecuencia de un cambio
    - 2.3.4. Responsable de medir las actividades vinculadas a cambios
  - 2.4. El grupo debe tener autoridad para decidir.
  - 2.5. Debería conformarse antes de que surjan los cambios y de ser necesario incorporar miembros para atender modificaciones específicas.
3. Seleccionar herramientas para la Gestión de Cambios del proyecto.
- 3.1. Una herramienta para la administración de requerimientos ofrece algunos atributos y permite que el analista agregue otros.
  - 3.2. La herramienta brinda también opciones de consulta en base a los valores de los atributos.
  - 3.3. Atributos de los requerimientos
    - 3.3.1. Identificador
    - 3.3.2. Estado
      - 3.3.2.1. Propuesto: El analista capturó el requerimiento de un usuario autorizado
      - 3.3.2.2. Analizado: El analista analizó ambigüedad, consistencia, precisión, prioridad
      - 3.3.2.3. Borrador: El analista esbozó una primera especificación

- 3.3.2.4. Aprobado: El requerimiento fue validado e incluido en el acuerdo base. Los usuarios lo consideran relevante y el equipo lo encuentra factible.
- 3.3.2.5. Implementado: El código está escrito y testeado a nivel unidad
- 3.3.2.6. Verificado: El requerimiento satisface el criterio de aceptación
- 3.3.2.7. Postergado: El requerimiento había sido aprobado pero se postergó para otra entrega
- 3.3.2.8. Eliminado: El requerimiento había sido aprobado pero se lo eliminó del acuerdo base
- 3.3.2.9. Rechazado: El requerimiento se propuso pero no se aprobó.
- 3.3.2.10. Requerimientos del Usuario
  - 3.3.2.10.1. Especificado: La especificación fue validada con el usuario y alineada con los RN y se le asignó una prioridad
  - 3.3.2.10.2. Derivado: El analista derivó todos los requerimientos funcionales y para los datos que surgen del RU
- 3.3.2.11. Requerimientos Funcionales
  - 3.3.2.11.1. Especificado: El requerimiento está especificado, se puede rastrear su origen y verificar su precisión y consistencia. El modelo de datos asociado al RF está completo.
- 3.3.3. Racionalidad
  - 3.3.3.1. Justifica por qué el requerimiento es relevante para el producto.
  - 3.3.3.2. La racionalidad va a ser útil para asignar prioridades y por lo tanto para replanificar o recortar requerimientos cuando surgen cambios y se agregan nuevos requerimientos.
  - 3.3.3.3. Un nuevo requerimiento puede provocar también que otro anterior deje de tener justificación y por lo tanto pueda eliminarse.
- 3.3.4. Descripción
- 3.3.5. Contexto
- 3.3.6. Antecedentes
- 3.3.7. Prioridad
- 3.3.8. Método de validación del criterio de aceptación
- 3.3.9. Fecha de creación y de la última modificación
- 3.3.10. Identificador de la última versión
- 3.3.11. Identificador de iteración o entrega asignada
- 3.3.12. Autor del requerimiento y responsable de la última versión
- 3.3.13. Origen o fuente de requerimiento
- 3.3.14. Participantes a contactar cuando surjan preguntas o se requieran decisiones

#### 4. Definir un procedimiento de gestión de cambios.

Cuando el cambio se acepta la documentación debe ser modificada, dando lugar a una nueva versión, que refleje los cambios. Ni el desarrollador, ni el diseñador deberían realizar cambios que el analista no ha incluido en el SRS. La verificación se realiza siempre en base a la documentación, que se asume vigente.

## Seguimiento de estado

- Definir el conjunto de estados posibles
  - Pendientes: Durante el desarrollo de un proyecto surge la necesidad de responder numerosas preguntas, resolver conflictos y tomar decisiones. En ocasiones puede hacerse en el momento que surge la necesidad pero en muchos casos es necesario esperar y es fundamental mantener registro de las cuestiones que quedan pendientes.
- Registrar el estado de cada requerimiento
- Rastrear la distribución de estados

## Gestión de versiones

El control de versiones está ligado a la gestión de cambios ya que permite hacer un seguimiento histórico de cambios .

Cada miembro del equipo de desarrollo debe tener acceso a la última versión de cada requerimiento. Los cambios deben quedar documentados y ser comunicados a todos los afectados.

### Objetivos

- Retener la historia de los cambios.
- Rastrear los cambios a través de las versiones de requerimientos o conjuntos de requerimientos
- Permite deshacer cambios para restablecer un ítem que ha sido modificado.

### Estrategias

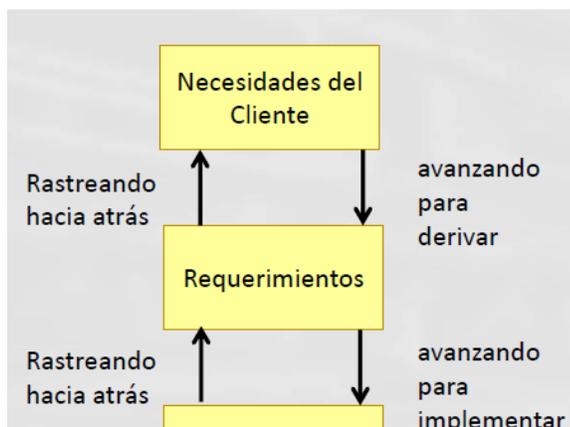
- Herramienta específica para AR: Las herramientas específicas permiten identificar la versión de cada requerimiento y de cada conjunto de requerimientos
  - identificador a la versión
  - descripción el cambio
  - fecha
  - responsable
- Procesador de textos con mecanismo de revisión: Revisión automática del procesador de textos: inserta marcas hasta que se llega al acuerdo base y se aceptan las revisiones.
- Etiquetado manual: Mecanismo manual de etiquetado: v1.1 v1.2 las fechas confunden

## Trazabilidad

Rastreo de requerimientos con sus correspondientes piezas de diseño, con código fuente, con test de verificación.

La calidad de los requerimientos está fuertemente ligada a la documentación de las dependencias y enlaces lógicos entre los requerimientos individuales y otros elementos del sistemas

Para que un requerimiento sea rastreable debe tener una etiqueta persistente que lo identifique unívocamente



## **Componentes**

- otros requerimientos
- reglas del negocio
- piezas de diseño
- módulos de código
- componentes de testing
- lecciones de los manuales de ayuda

## **Objetivos**

- Encontrar requerimientos ausentes
- Encontrar requerimientos innecesarios
- Evaluar el impacto de los cambios
- Verificar componentes de código
- Certificar reglamentaciones o estándar
- Reemplazar componentes
- Reusar componentes

## **Cardinalidad en los enlaces**

One to one: Una pieza de diseño se corresponde a un único módulo de código

One to many: Un RF es verificado por varios casos de prueba

Many to many: Un CU se corresponde a varios RF y un RF puede corresponder a varios CU

Definir relaciones y dependencias entre requerimientos

Negociar cambios en el plan de proyecto considerando la evaluación del impacto y ajustar el plan de proyecto de acuerdo a lo negociado

Anticipar evolución o cambios de requerimientos para minimizar el impacto

Acuerdo base: establece una línea que separa el desarrollo de la administración de requerimientos

Se establece cuando termina el desarrollo de requerimientos. Separa el desarrollo de requerimientos de la gestión de cambios.

Permite evaluar el impacto de los cambios propuestos para los requerimientos e incorporar los cambios aprobados en el proyecto de manera controlada.

Cualquier cambio posterior al acuerdo base que afecte al acuerdo base va a implicar renegociar el plan del proyecto.

## Entregables

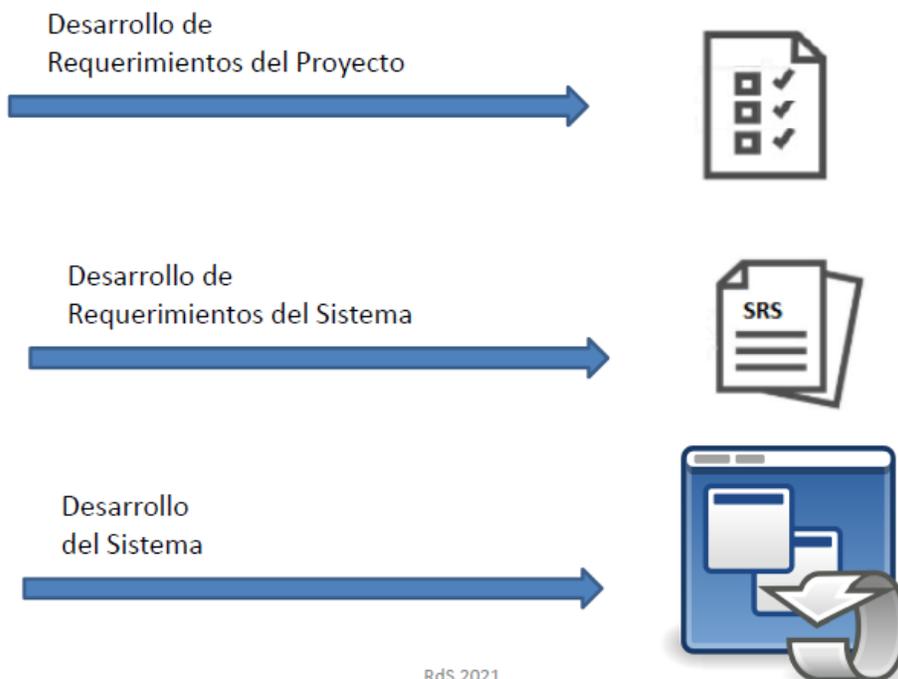
1. Estudio de factibilidad
2. Proceso de desarrollo de requerimientos del proyecto, cuyo resultado es un plan del proyecto (entregable)
3. Proceso de desarrollo de requerimientos del sistema, cuyo resultado es un documento de especificación de requerimientos (entregable).

Los destinatarios de este documento son:

- Clientes y usuarios
- Diseñadores
- Desarrolladores / Implementadores
- Testers

Este documento de especificación de requerimientos será una herramienta valiosa durante todo el ciclo del sistema. Debe mantenerse actualizado.

4. Proceso de desarrollo del sistema, cuyo resultado es un sistema o producto que incluye una o más componentes de sw (entregable)



## Requerimientos y diseño

A partir de los requerimientos del negocio, del usuario y de las reglas del negocio, el analista deriva los requerimientos funcionales y de los datos.

El diseño de la arquitectura del sistema parte del conjunto de requerimientos funcionales y de datos, e incluye también a los atributos de calidad y a las restricciones (requerimientos no funcionales)

Aunque la metodología afecta fundamentalmente al diseño e implementación del sistema, durante el desarrollo de requerimientos podemos elegir herramientas consistentes con la metodología que se va a utilizar después.

## Requerimientos y testing

La verificación de requerimientos está muy ligada a la validación del producto.

Durante la verificación, el responsable de testing controla que la implementación corresponda a los requerimientos especificados.

Si los requerimientos satisfacen los atributos de la verificación, el documento de especificación de requerimientos (srs) define los criterios de aceptación que van a permitir decidir si el sistema satisface los requerimientos.

La verificación implica revisar cada componente, subsistema, y el sistema global, con relación a los requerimientos.

Durante la validación, los clientes y usuarios controlan si el producto satisface sus necesidades en términos de correctitud, completitud y consistencia.

## Éxito de un proyecto

El proyecto se completa en el tiempo previsto, con el costo presupuestado y con los recursos asignados.

El producto desarrollado satisface las expectativas de clientes y usuarios.

El éxito del proyecto quedará determinado por la calidad del producto, la cual está íntimamente relacionada con el cumplimiento de los objetivos del negocio, los cuáles guían el desarrollo de los requerimientos del producto.

## Proceso de desarrollo

### Roles

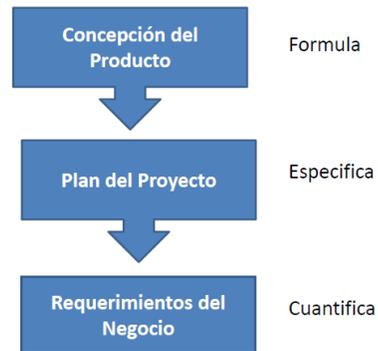
#### Administrador del proyecto

1. Participa en el estudio de factibilidad: es una actividad breve que se realiza antes de comenzar el desarrollo. La meta es establecer por qué y para qué se va a desarrollar el producto. El administrador establece la factibilidad técnica y económica del proyecto.

2. Elabora el plan de proyecto, que incluye los requerimientos del proyecto. En este plan, el administrador del proyecto especifica:

a. Objetivo del proyecto: el proyecto se formula y se ejecuta para satisfacerlo. El objetivo surge de un problema, y se enmarca en el objetivo de la organización.

El objetivo del proyecto se formula durante la concepción del producto, se especifica en el plan del proyecto y se refina en los requerimientos del negocio, cuantificando el beneficio de una manera medible: el objetivo del negocio.



b. Alcance del proyecto

c. Criterios de calidad que deben satisfacerse

d. Recursos disponibles (recursos humanos, infraestructura: todo lo que el equipo de desarrollo necesita, como hw, equipamiento, laboratorios, salas, etc).

e. Presupuesto

f. Cronograma

g. Riesgos

El alcance y la calidad, van a estar ligados a la disponibilidad de recursos, el tiempo y el costo.



Un proyecto en general estará dividido en subproyectos, y cada uno tendrá su propio alcance, presupuesto y cronograma, y criterios para definir si se ha terminado y medir su éxito.

Durante el desarrollo de cada subproyecto se generan 1 o más entregables. Las etapas pueden ser:

1. Secuenciales
2. Paralelas
3. Iterativas Incrementales

## Procesos, productos, innovación y tecnología (DR12)

Cómo el desarrollo de nuevos productos puede impactar en los procesos

