



## Bloque II

### *Tema 4*

### *Análisis de Requisitos*



### Contenido

- Definición de Requisitos
- Tipos de Requisitos
- Gestión de Requisitos
- Ingeniería de Requisitos
- Acciones a Realizar en la Especificación de Requisitos
  - Extracción de Requisitos
  - Análisis de Requisitos
  - Especificación de Requisitos
  - Validación de Requisitos
- Especificación de requisitos del Software (ERS)
- Especificación de requisitos Ágiles



# Definición de Requisitos

## *¿Qué son los Requisitos?*

- Describen los servicios que debe proporcionar el sistema y sus restricciones operativas
- Un requisito puede ser una simple declaración abstracta de alto nivel o bien una definición detallada y formal de una función del sistema
- Es necesario hacer una separación entre niveles de descripción:
  - Requisitos de Usuario
  - Requisitos del Sistema



### Tipos de Requisitos

- Requisitos del Usuario (*descripción de alto nivel*)
- Requisitos del Sistema (*descripción detallada del sistema*)
  - Requisitos Funcionales
  - Requisitos No Funcionales:
    - ✓ Requisitos Del Producto
    - ✓ Requisitos Organizacionales
    - ✓ Requisitos Externos
  - De Dominio



### Tipos de Requisitos

- Requisitos del Usuario:
  - Descripciones, en lenguaje natural o diagramas, de lo que se espera que el sistema proporcione y las restricciones bajo las cuales debe funcionar
- Requisitos del Sistema:
  - Establecen con detalle las funciones, servicios y restricciones operativas del sistema
    - ✓ Deben ser precisos
    - ✓ Definir exactamente qué es lo que se va a implementar
    - ✓ Puede ser parte del contrato entre el comprador y el desarrollador

# Tipos de Requisitos

### Definición del requerimiento del usuario

1. LIBSYS controlará todos los datos requeridos por las agencias que licencian los derechos de autor en el Reino Unido y en otra parte.

### Especificación de los requerimientos del sistema

- 1.1 Al hacer una petición de un documento del LIBSYS, el solicitante se presentará con un formulario que registre los detalles del usuario y de la petición hecha.
- 1.2 El formulario de petición del LIBSYS será almacenado en el sistema durante cinco años desde la fecha de la petición.
- 1.3 Todos los formularios de peticiones del LIBSYS se deben indexar por usuario, por el nombre del material solicitado y por el proveedor de la petición.
- 1.4 El LIBSYS mantendrá un fichero en el que se registrarán todas las peticiones que se han hecho al sistema.
- 1.5 Para el material donde se aplican los derechos de préstamo del autor, los detalles del préstamo serán enviados mensualmente a las agencias que licencian los derechos de autor que se han registrado con LIBYSS.



### Tipos de Requisitos

- **Requisitos Funcionales**

- Son declaraciones de los servicios que debe proporcionar el sistema
- Especifica la manera en que éste debe reaccionar a determinadas entradas
- Especifica cómo debe comportarse el sistema en situaciones particulares
- Pueden declarar explícitamente lo que el sistema no debe hacer

#### Ejemplo Requisitos del Sistema

##### Ejemplo LIBSYS

Un sistema que proporciona un interfaz común para el acceso a distintas bibliotecas

Los usuarios pueden buscar, descargar e imprimir los artículos para uso personal

#### Ejemplo Requisitos Funcionales

1. El usuario deberá tener la posibilidad de buscar en el conjunto inicial de la base de datos o seleccionar un subconjunto de ella.
2. El sistema deberá proporcionar visores adecuados para que el usuario lea documentos en el almacén de documentos.
3. A cada pedido se le deberá asignar un identificador único (ID\_PEDIDO), que el usuario podrá copiar al área de almacenamiento permanente de la cuenta.



### Tipos de Requisitos

- Si los requisitos no están bien definidos, pueden surgir problemas
- Requisitos ambiguos dan lugar a diferentes interpretaciones por parte de los usuarios y los desarrolladores
- Considérese el término **visores apropiados**
  - Intención del Usuario – visores especiales para cada tipo de documento
  - Interpretación del Desarrollador – proporcionar un visor de texto que muestre el contenido del documento
- En principio, lo deseable es que los requisitos fueran completos y consistentes
  - Completos: Recogen la descripción de todos los servicios esperados por el usuario
  - Consistentes: No hay contradicciones entre estas especificaciones
- En la práctica es imposible producir un documento de requisitos completo y consistente





### Tipos de Requisitos

#### Requisitos No Funcionales

- No se refieren a funciones específicas que proporciona el sistema
- Son restricciones de los servicios o funciones ofrecidas por el sistema (fiabilidad, tiempo de respuestas, capacidad de almacenamiento, etc.)
- Generalmente se aplican al sistema en su totalidad
- Surgen de las necesidades del usuario (restricciones de presupuesto, políticas de la organización, necesidad de interoperabilidad, etc.)
- Definen propiedades y restricciones del sistema: tiempos de respuesta, requisitos de almacenamiento etc.
- También puede referirse al proceso de desarrollo: uso de una herramienta CASE, lenguaje de programación o metodología concreta
- En general, suelen ser más críticos que los requisitos funcionales. Si no son satisfechos, el sistema no resulta útil

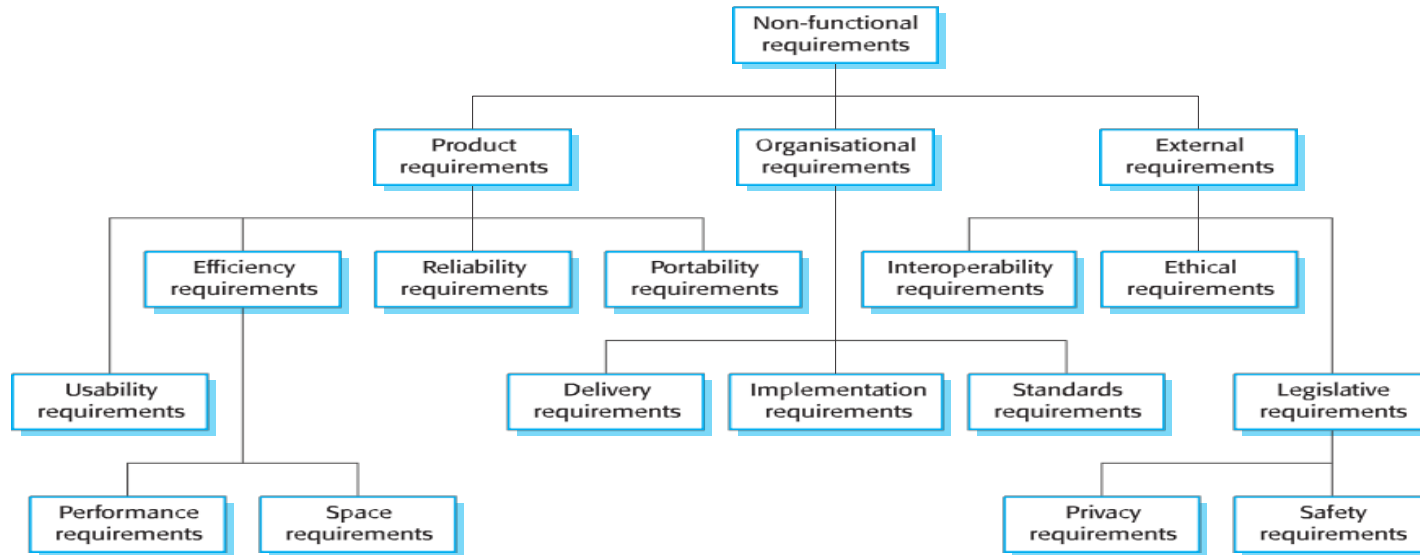


# Tipos de Requisitos

## Tipos de Requisitos No Funcionales

- Requisitos Del Producto:
  - Especifican el comportamiento del producto.
  - *Ejemplos: rapidez de la ejecución, capacidad de memoria, fiabilidad, etc.*
- Requisitos Organizacionales:
  - Derivan de políticas y procedimientos existentes en la organización del cliente y del desarrollador.
  - *Ejemplos: Estándares de procesos, métodos de diseño, lenguajes de programación, métodos de entrega, etc.*
- Requisitos Externos:
  - Se derivan de factores externos al sistema y a su procesos de desarrollo.
  - *Ejemplos: Requisitos de interoperabilidad, legislativos, éticos, etc.*

## Tipos de Requisitos



### Requerimiento del producto

8.1 La interfaz de usuario del LIBSYS se implementará como HTML simple sin marcos o applets Java.

### Requerimiento organizacional

9.3.2 El proceso de desarrollo del sistema y los documentos a entregar deberán ajustarse al proceso y a los productos a entregar definidos en XYZCo-SP-STAN-95.

### Requerimiento externo

10.6 El sistema no deberá revelar al personal de la biblioteca que lo utilice ninguna información personal de los usuarios del sistema aparte de su nombre y número de referencia de la biblioteca.



# Tipos de Requisitos

### Metas y Requisitos

- Puede ser complicado de precisar cuando se trabaja con requisitos no funcionales
- Requisitos imprecisos → Difíciles de verificar
- Meta
  - Un propósito general del usuario, como *facilidad de uso*
- Requisito no funcional verificable
  - Una frase que implica alguna medida objetiva
- Las metas son útiles para los desarrolladores porque transmiten los deseos del usuario del sistema
- Una Meta
  - El sistema debe ser fácil de usar por parte de usuarios cualificados y organizado de tal forma que los posibles errores del usuario se minimicen
- Un Requisito No Funcional verificable:
  - Los usuarios cualificados deben estar en condiciones de utilizar todas las funciones del sistema después de 2 horas de entrenamiento. La media de errores por usuario no deberá superar los 2 errores/día después de dicho entrenamiento



# Tipos de Requisitos

## Requisitos de Dominio

- Proviene del dominio de aplicación del sistema y no de las necesidades del usuario
- Reflejan características y restricciones del dominio de la aplicación
- Pueden ser nuevos requisitos funcionales, restricciones sobre requisitos funcionales que ya existían o definir cómo deben llevarse a cabo ciertas tareas
- Ejemplo de requisito de dominio de LYBSYS
  1. Deberá existir una interfaz de usuario estándar para todas las bases de datos que estará basada en el estándar Z39.50.
  2. Debido a las restricciones en los derechos de autor, algunos documentos deberán borrarse inmediatamente después de su llegada. Dependiendo de los requerimientos del usuario, estos documentos se imprimirán de forma local en el servidor del sistema para ser distribuidos de forma manual al usuario o se enviarán a la impresora de la red.
- Comprensión: Los requisitos se expresan en términos del dominio de la aplicación. No siempre entendibles por los desarrolladores
- Conocimiento Implícito: Los especialistas del dominio conocen el área y por tanto no piensan en hacer los requisitos más explícitos



### Tipos de Requisitos

#### Requisitos de Usuario

- Deben describir requisitos funcionales y no funcionales de modo que sean entendibles por usuarios del sistema que carecen de conocimientos técnicos
- Se describen usando lenguaje natural, tablas y diagramas
- Problemas : Falta de Claridad
- Es difícil ser preciso sin complicar la lectura del documento
- Mezclar tipos de requisitos: Se acostumbra a mezclar los requisitos funcionales y no funcionales
- Fusión de Requisitos: Se agrupan varios requisitos en la definición de uno solo
- Ejemplo: El sistema LIBSYS debe proporcionar un subsistema de contabilidad que mantenga registros de todos los pagos hechos por los usuarios. Los responsables deben poder configurar este subsistema para poder ofrecer descuentos a los usuarios habituales.



### Tipos de Requisitos

#### Consejos para la definición de requisitos

- Idear un formato (+ ó –) estándar y ceñirse a él para todos los requisitos
- Utilizar el lenguaje de forma consistente:
  - Presente para requisitos obligatorios
  - Condicional para requisitos deseables
- Utilizar el subrayado para resaltar las partes relevantes
- Evitar el uso de jerga

#### Requisitos de interfaz (Interface requirements)

- Describen el formato con el que la aplicación se comunica con su entorno
- Ejemplo (comunicación con usuarios): El coste del envío se mostrará en la esquina inferior derecha
- Ejemplo (comunicación con otras aplicaciones): Los pedidos se transmitirán en formato XML utilizando la plantilla DTD detallada en el anexo IV





### Gestión de Requisitos

- Tras la identificación de los requisitos, éstos deben ser:
  - Incluidos en un **Catálogo**
  - **Analizados**, con el objetivo de detectar inconsistencias, ambigüedades, duplicidad o escasez de información
  - Validados por los clientes/usuarios
- Además se deben realizar actividades que confirmen que los requisitos **se cumplen** (trazabilidad y coherencia)
- Las características que debe tener un requisito son: **Identificador**, Autor, Tipo de requisito, **Descripción**, **Prioridad (Alta/media/baja)**, Estado (propuesto/aprobado/incorporado) y Fecha de Creación/Revisión
- Se deben organizar:
  - Por Subsistemas
  - Por tipo (RFX / RNFX)
  - Jerárquicamente (RF1 / RF1.1 / RF1.1.1)





# Ingeniería de Requisitos

- La ingeniería de requisitos del software es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refinan en detalle los requisitos del sistema y el papel asignado al software
- Tanto el desarrollador como el cliente tienen un papel activo en la ingeniería de requisitos que es un conjunto de actividades que son denominadas análisis. El cliente intenta replantear un sistema confuso, a nivel de descripción de datos, funciones y comportamiento, en detalles concretos. El desarrollador actúa como interrogador, como consultor, como persona que resuelve problemas y como negociador
- El análisis y la especificación de requisitos pueden parecer una tarea relativamente sencilla, pero las apariencias engañan. El contenido de comunicación es muy denso. Abundan las ocasiones para malas interpretaciones o falta de información. Es muy probable que haya ambigüedad. El dilema al que se enfrenta el ingeniero de software puede entenderse muy bien repitiendo la famosa frase de un cliente anónimo: "Sé que cree que entendió lo que piensa que dije, pero no estoy seguro de que se dé cuenta de que lo que escuchó no es lo que yo quise decir"



# Ingeniería de Requisitos

- El análisis de requisitos es una tarea de ingeniería del software que cubre el hueco entre la definición del software a nivel sistema y el diseño de software. El análisis de requerimientos permite al ingeniero de sistemas especificar las características operacionales del software (función, datos y rendimientos), indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe cumplir el software



# Acciones a Realizar en la Especificación de Requisitos

Las principales actividades que la componen son:

- **Extracción de requisitos**
  - Proceso por el cual los clientes descubren, revelan y comprenden los requisitos que desean
- **Análisis de requisitos**
  - Proceso de razonamiento sobre los requisitos obtenidos en la etapa anterior, detectando y resolviendo posibles inconsistencias o conflictos, coordinando los requisitos relacionados
- **Especificación de requisitos**
  - Proceso de redacción o registro de los requisitos. Para este proceso puede recurrirse al lenguaje natural, formal, modelos, gráficos etc.
- **Validación de requisitos**
  - Proceso de confirmación por parte de los usuarios/clientes de los requisitos especificados en el que se comprueba su validez, consistencia, compleción (completitud) etc.

La realización de las mismas suele apoyarse en distintas técnicas. Así para la extracción de los requisitos se emplean **técnicas de recogida de información como la observación, entrevistas, grupos de trabajo**. Para el análisis y especificación existen técnicas gráficas como los **Diagramas de flujo de datos y los casos de uso**

Para la validación suele recurrirse **listas de comprobación junto con técnicas de revisión y comprobación**



# Acciones a Realizar en la Especificación de Requisitos

Principales técnicas de captura y análisis de requisitos:

- Entrevistas
- Desarrollo conjunto de aplicaciones (JAD)
- Prototipado
- Observación
- Estudio de documentación
- Cuestionarios
- Tormenta de ideas (*Brainstorming*)
- ETHIC (Effective Technical and Human Implementation of Computer-based System)



# Especificación de Requisitos

- El objetivo de la Especificación de Requisitos es obtener un documento de especificación de requisitos del Software ([ERS](#))
- Documento que define, de forma completa, precisa y verificable los requisitos que debe cumplir el sistema tanto funcionales como no funcionales así como las restricciones aplicables al diseño (software y hardware) [[IEEE/ANSI 830-1998](#)] su contenido es:

### 1. Introducción

1.1. Propósito

1.2. Ámbito del Sistema

1.3. Definiciones, Acrónimos y Abreviaturas

1.4. Referencias

1.5. Visión General del Documento



# Especificación de Requisitos

## 2. Descripción General

- 2.1. Perspectiva del Producto
- 2.2. Funciones del Producto
- 2.3. Características de los Usuarios
- 2.4. Restricciones
- 2.5. Suposiciones y Dependencias
- 2.6. Requisitos Futuros

## 3. Requisitos Específicos

- 3.1. Interfaces Externas
- 3.2. Funciones
- 3.3. Requisitos de Rendimiento
- 3.4. Restricciones de Diseño
- 3.5. Atributos del Sistema
- 3.6. Otros Requisitos

## 4. Apéndices



# Especificación de Requisitos

Características deseable en una BUENA especificación de requisitos

- No ambigua
- Completa
- Fácil de verificar
- Consistente (coherente)
- Clasificada por importancia o estabilidad
- Fácil de modificar
- Fácil identificación del origen y de las consecuencias de cada requisito
- De fácil utilización durante la fase de explotación y de mantenimiento



# Especificación de Requisitos

### Especificación de los requisitos de las interfaces

- Las interfaces externas del software juegan un papel crucial en el propio software puesto que es lo que el usuario percibe más fácilmente y donde más influyen sus preferencias
- Las interfaces coinciden con las tradicionalmente llamadas Entradas/ Salidas del sistema, haciendo referencia por un lado a pantallas de introducción de datos, sensores, ficheros, etc. como entradas y por otro a diálogos de información, listados o salidas en papel, ficheros etc. como salidas de datos
- El objetivo de la definición de las interfaces de E/S es la estabilidad del modo en que el sistema va a interactuar con el exterior del mismo, siendo fundamental no sólo contar con la opinión de los usuarios, sino con criterios que le ayuden a decidir qué es mejor para su trabajo





# Especificación de Requisitos Ágiles

## Historias de Usuario

- Las historias de usuario son utilizadas en los métodos ágiles para la especificación de requisitos, son una descripción breve de una funcionalidad software tal y como la percibe el usuario (Mike Cohn, 2004)
- Describen lo que el cliente o el usuario quiere que se implemente y se escriben con una o dos frases utilizando el lenguaje común del usuario. El pensamiento de las personas se estructura siguiendo una narrativa, una historia, así es como entendemos el mundo
- Cada historia de usuario debe ser limitada, esta debería poderse memorizar fácilmente y escribir sobre una tarjeta o post-it (card)
- Poco antes de ser implementadas, estas van acompañadas de conversaciones con los usuarios y la definición de los criterios de validación asociados



# Especificación de Requisitos Ágiles

## Epic, Temas y Tareas

- Se denomina epic a una superhistoria de usuario que se distingue por su gran tamaño, a diferencia de las historias de usuario, que tienen baja granularidad, los epics tienen una alta granularidad
- Es una etiqueta que aplicamos a una historia grande, cuyo esfuerzo es demasiado grande para completarla de una sola vez o en un solo sprint
- Los epics suelen tener un flujo asociado por el cual se puede dividir en historias de usuario, en otras palabras, las historias de usuario resultantes de la descomposición de un epic están íntimamente relacionadas entre sí
- A medida que aumenta su prioridad y se acerca al momento de su implementación, el equipo la descompone en historias de usuario con un tamaño más adecuado para ser gestionada con los principios y técnicas ágiles: estimación y seguimiento cercano (normalmente diario).



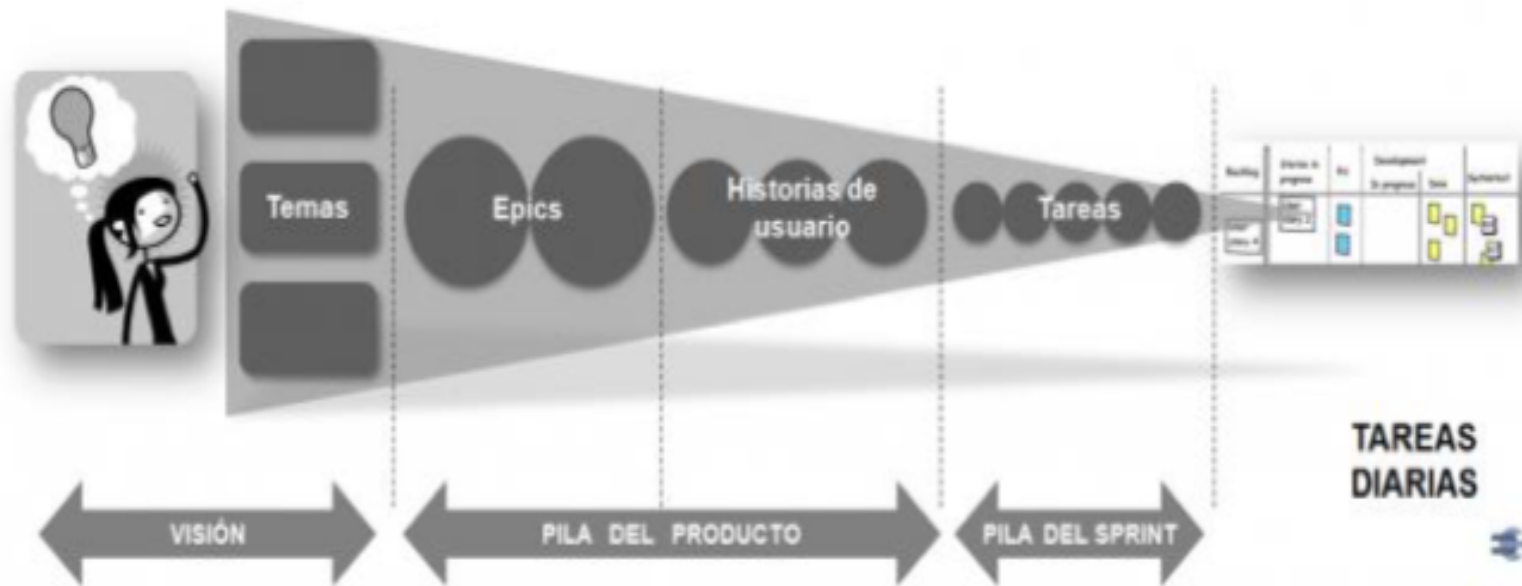
# Especificación de Requisitos Ágiles

### Epics, Temas y Tareas (cont.)

- Por encima de los epics se encuentran los temas que representan a una colección de epics y/o historias de usuario relacionados para describir un sistema o subsistema en su totalidad.
  - Por ejemplo en un sistema de software para gestión contable, el conjunto de epics: “Altas, bajas y mantenimiento de clientes”, “Facturaciones puntuales y recurrentes”, “Consultas de navegación y acciones de fidelización”, “Pedidos”, “Devoluciones” se podrían denominar como el tema de la gestión de clientes.
- Por debajo de las historias de usuario se encuentran las tareas. Estas son resultado de la descomposición por parte del equipo de las historias de usuario en unidades de trabajo adecuadas para gestionar y seguir el avance de su ejecución

## Especificación de Requisitos Ágiles

Epics, Temas y Tareas (cont.)





# Especificación de Requisitos Ágiles

### Epics, Temas y Tareas (cont.)

- En scrum, y en las metodologías ágiles en general, una pila del producto puede contener tanto historias de usuario como epics. La pila del producto debe de estar ordenada por prioridad y el nivel de detalle de cada elemento debe de ir en relación a la posición del mismo dentro de la pila
- En una lista larga, algo muy poco prioritario no tiene sentido tenerlo al detalle, porque probablemente cambiará a lo largo del proyecto, e incluso puede que ni se desarrolle
- De la mitad hacia el final de la lista, donde está lo menos prioritario, es el lugar de los epics
- A medida que nos acercamos a los elementos más prioritarios, el detalle debe aumentar, por tanto las historias de usuario son los elementos que deben de encabezar la lista



# Especificación de Requisitos Ágiles

## Información en una Historia de Usuario

- Es preferible no adoptar formatos regidos, si no adaptarlos a las características de la empresa y del proyecto
- Por ello aparte de los 4 campos que se consideran necesarios, se puede incluir cualquier campo que se considere útil para el proyecto
- Estos campos son:
  - **ID:** identificador de la historia de usuario, único para la funcionalidad o trabajo
  - **Descripción:** el estilo puede ser libre, aunque debe responder a tres preguntas: ¿quién se beneficia?, ¿qué se quiere? y ¿cuál es el beneficio?, se recomienda seguir el siguiente patron:

Como [rol del usuario], quiero [objetivo], para poder



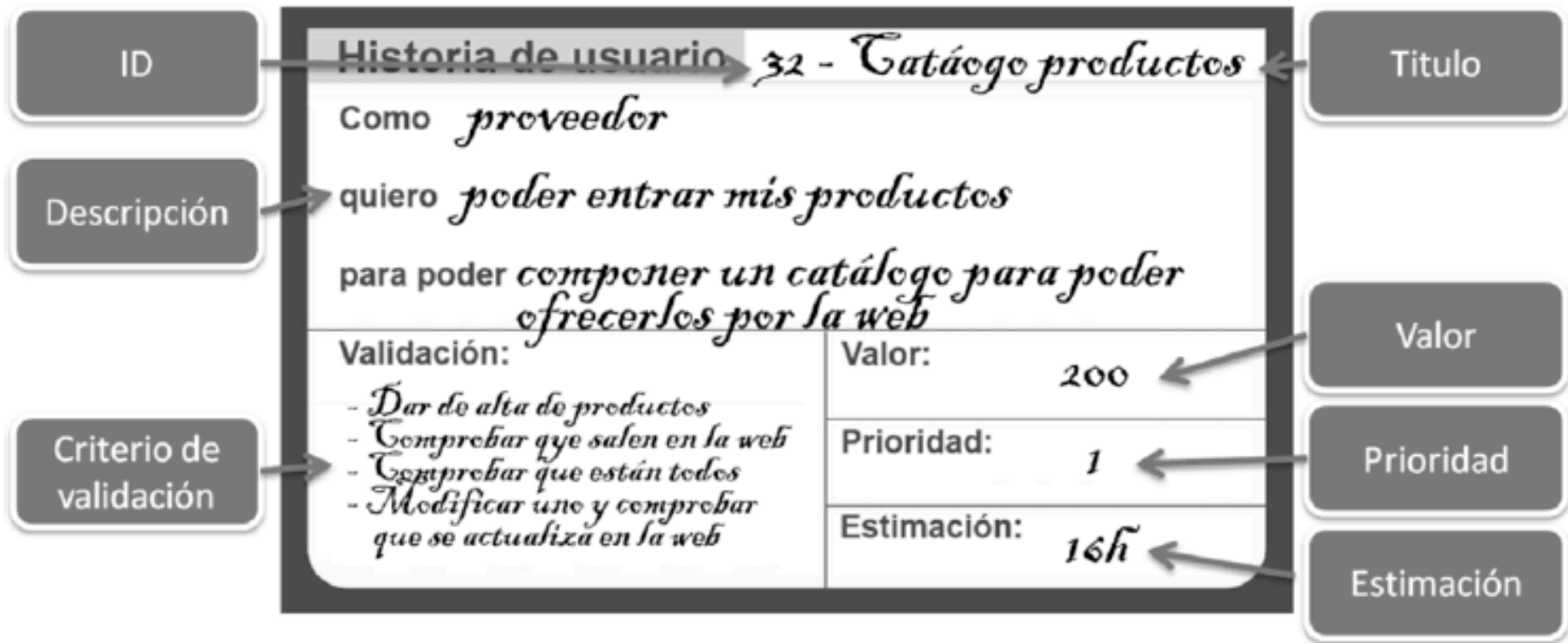
# Especificación de Requisitos Ágiles

### Información en una Historia de Usuario (cont.)

- **Estimación:** será el esfuerzo necesario en tiempo ideal de implementación de la historia de usuario, se pueden utilizar los puntos de historia expresados en tiempo teórico de desarrollo/persona
- **Prioridad:** nos permite determinar el orden en el que las historias de usuario deben ser implementadas
- Deponiendo del tipo de proyecto, funcionamiento del equipo y la organización pueden ser aconsejables otros campos, como:
  - Título
  - Criterio de Validación, pruebas de aceptación consensuadas con el cliente o usuario
  - Valor, normalmente numérico, este campo junto con otros nos dará la priorización
  - Dependencias
  - Etc....

## Especificación de Requisitos Ágiles

Información en una Historia de Usuario (cont.)







## Tema 4.- Análisis de Requisitos

# Especificación de Requisitos Ágiles

### Información en una Historia de Usuario (cont.)

Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre historia: Cambiar dirección de envío	
Prioridad en negocio: Alta	Riesgo en desarrollo: Bajo
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: José Pérez	
<b>Descripción:</b> Como cliente quiero cambiar la dirección de envío de un pedido para que me pueda llegar a casa o a la oficina	
<b>Validación:</b> El cliente puede cambiar la dirección de entrega de cualquiera de los pedidos que tiene pendiente de envío	