

Unidad 2.1.

# CASOS DE USOS Y REQUISITOS



**UAIOnline**  
**Ultra**»»



# Casos de uso y Requisitos

## Unidad 2.1

- **OBJETIVOS**

- Comprender el concepto de Requisitos funcionales y el concepto de casos de uso y poder relacionarlos.



# Casos de uso y Requisitos

## Unidad 2.1

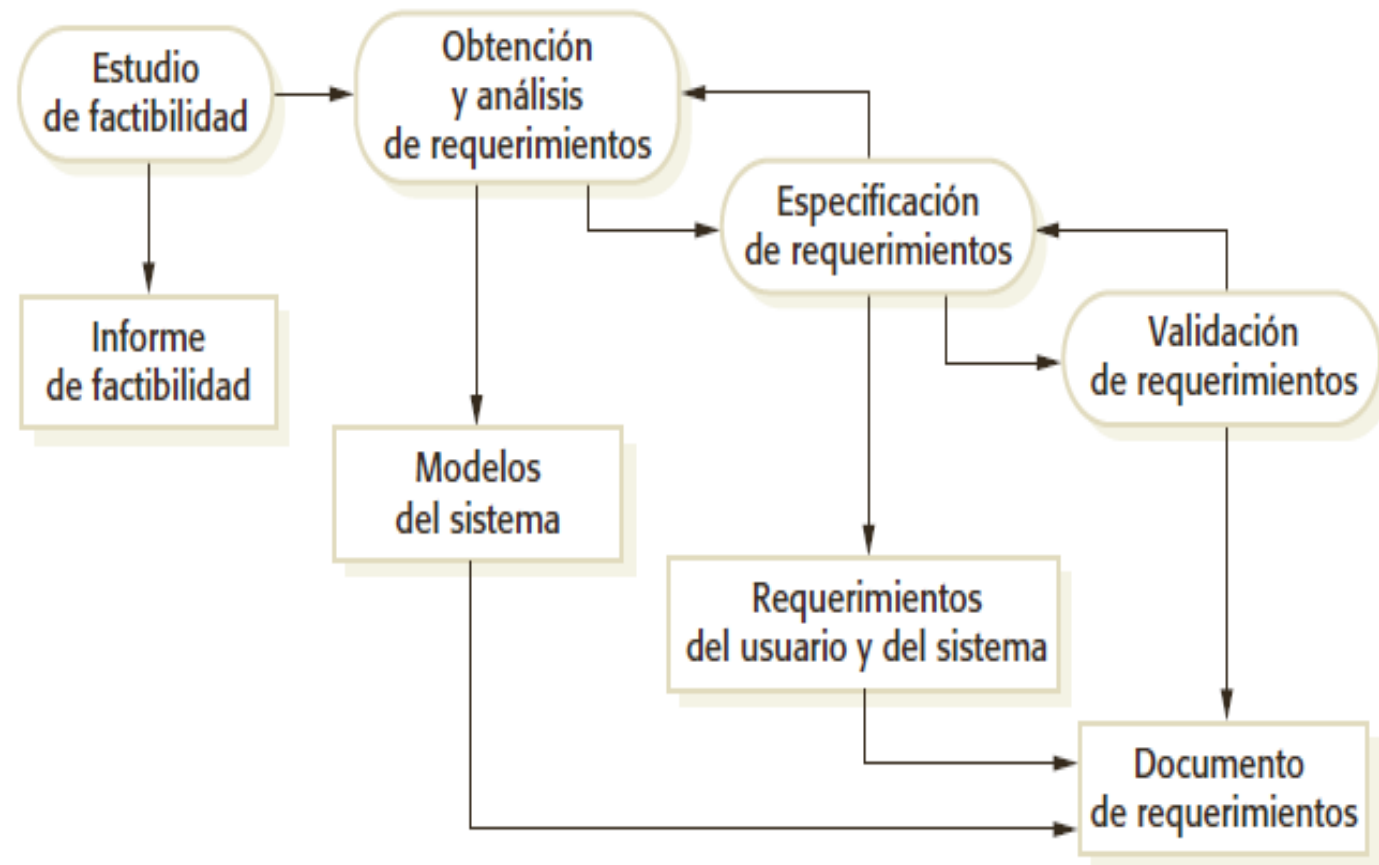
¿Qué es el software?



# INGENIERÍA DE REQUISITOS

- *La Ingeniería de Requisitos direcciona el proceso de elicitación (inducir), definición, modelado, análisis, especificación y validación de los requisitos de un sistema y de su software, basado en un enfoque sistemático, separando el "**qué**" del "**cómo**" del diseño.*

# EL PROCESO DE INGENIERÍA DE REQUISITOS



# ¿QUÉ ES UN REQUISITO?

- Flujo de trabajo fundamental cuyo proposito esencial es orientar el desarrollo hacia el sistema correcto.
- Esto se lleva a cabo mediante la descripción de los requisitos del sistema de forma tal que se pueda llegar a un acuerdo entre el cliente (usuario) y los desarrolladores del sistema, acerca de lo que el sistema debe hacer y lo que no.
- Entender en forma clara la necesidad del usuario.
- El conjunto de todos los requisistos forman la base para el desarrollo del sistema o el componente de Sistema.
- **Un requerimiento es una condición o capacidad a la que el sistema (siendo construido) debe conformar.**

# ¿QUÉ ES UN REQUISITO DE SOFTWARE?

- Una capacidad del software necesaria por el usuario para resolver un problema o alcanzar un objetivo, que sirva para la toma de decisiones.
- Una capacidad del software que debe ser reunida o poseída por un sistema o componente del sistema para satisfacer un contrato, especificación, estándar, u otra documentación formal.
- Es una descripción completa del comportamiento del sistema que se va a desarrollar.

# ESTRUCTURA DEL DOCUMENTO DE REQUERIMIENTOS - ERS

## ❑ Definición de Requerimientos No-funcionales.

- ❑ Definir las limitantes del sistema y del proceso de desarrollo.

## ❑ Evolución del Sistema.

- ❑ Definir las suposiciones fundamentales en las cuales el sistema se basa y se anticipan los cambios.

## ❑ Especificación de Requerimientos.

- ❑ Especificación detallada de los requerimientos funcionales del sistema.

## ❑ Apéndices.

- ❑ Descripción de la plataforma de Hardware del Sistema.
- ❑ Requerimientos de la base de Datos (quizá como un modelo ER)

## ❑ Índice.

Según ANSI/IEEE (Std. 830-1984), un documento de especificación de requisitos de software (ERS) debe ser: **No ambigua, Completa, Verificable, Consistente, Modificable, Trazable, Usable durante la operación y el mantenimiento.**



# CARACTERÍSTICAS DE UNA BUENA ERS/ I

Las características de una buena ERS son definidas por el estándar IEEE 830-1998. Una buena ERS debe ser:

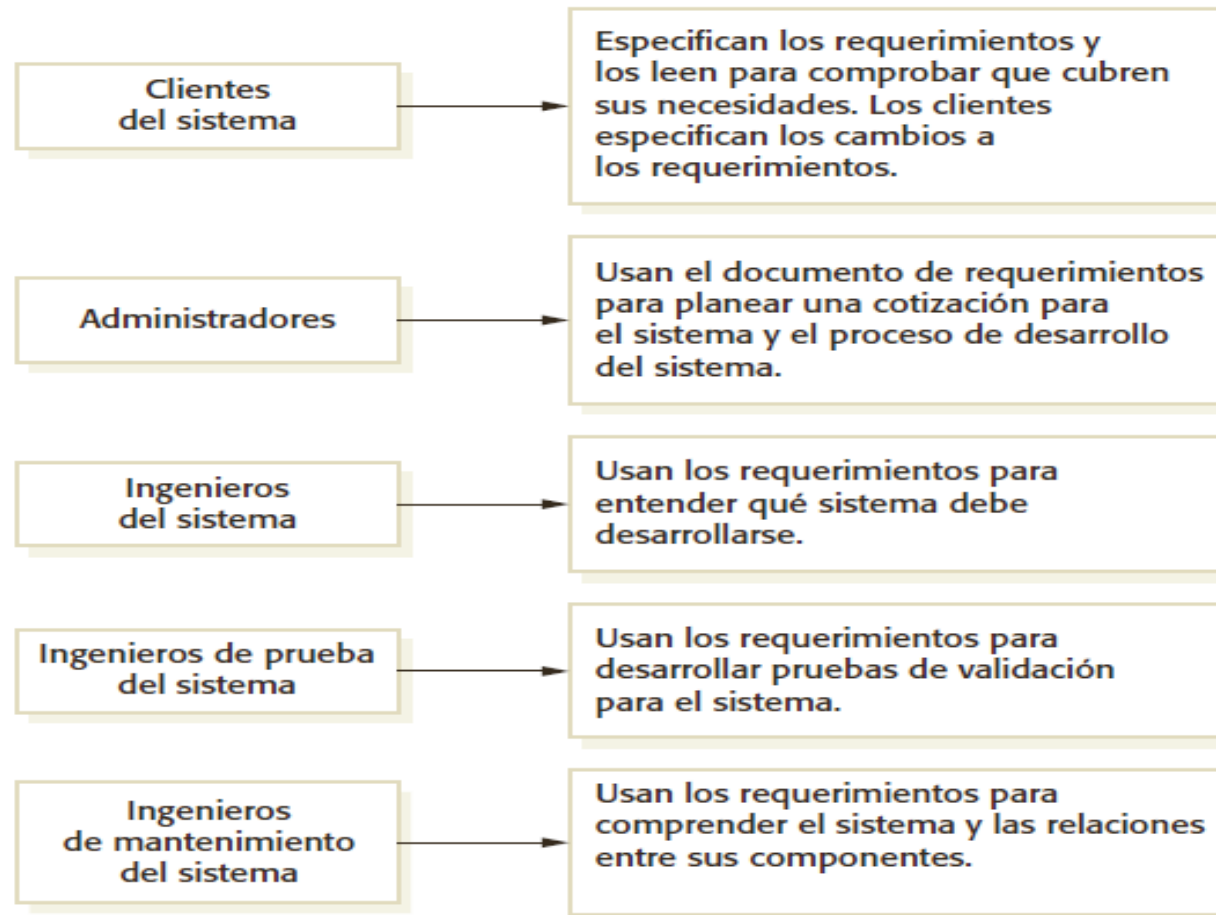
- **Completa**. Todos los requerimientos deben estar reflejados en ella y todas las referencias deben estar definidas.
- **Consistente**. Debe ser coherente con los propios requerimientos y también con otros documentos de especificación.
- **Inequívoca**. La redacción debe ser clara de modo que no se pueda mal interpretar.
- **Correcta**. El software debe cumplir con los requisitos de la especificación.

# CARACTERÍSTICAS DE UNA BUENA ERS/2

Las características de una buena ERS son definidas por el estándar IEEE 830-1998. Una buena ERS debe ser:

- **Trazable**. Se refiere a la posibilidad de verificar la historia, ubicación o aplicación de un ítem a través de su identificación almacenada y documentada.
- **Priorizable**. Los requisitos deben poder organizarse jerárquicamente según su relevancia para el negocio y clasificándolos en esenciales, condicionales y opcionales.
- **Modificable**. Aunque todo requerimiento es modificable, se refiere a que debe ser fácilmente modificable.
- **Verificable**. Debe existir un método finito sin costo para poder probarlo.

# USUARIOS DE UN DOCUMENTO DE REQUISITOS



# IDENTIFICACIÓN DE REQUISITOS

- Los Requerimientos toman vida desde que realizamos nuestro primer encuentro de interlocución con usuarios o clientes.
- Variedad de técnicas como entrevistas para intercambiar opiniones, brainstorming, prototipo, cuestionarios, etc.
- Cuando los requerimientos se logran redactar a un significativo nivel de detalle, tendremos listo el documento denominado “Especificación de Requerimientos”.
  - Definidos sin ambigüedad
  - Son completos
  - Tienen consistencia
  - Especifica el origen
  - Evita detalles de diseño
  - Están enumerados

# ELICITACIÓN

Es el proceso de adquirir (“eliciting”) [sonsacar] todo el conocimiento relevante necesario para producir un modelo de los requerimientos de un dominio de problema.

**Objetivo:** entender el dominio del problema en particular

¿Dónde encontrar el conocimiento?

**Problemas:**

Forma no utilizable del conocimiento

Dificultad cuando se trata de un experto humano

# TÉCNICA DE ELICITACIÓN

- Entrevista de comienzo y final abierto
- Entrevistas estructuradas
- Brainstorming (Lluvia de Ideas)
- Prototipos
- Escenarios
- Observación

**Posible tema de Investigación**

# CATEGORÍAS DE PROBLEMAS EN LA ELICITACIÓN DE LOS REQUISITOS

❑ Los problemas se pueden dividir en 3 grupos:

- **Problemas de Alcance:**

- Organización -> entender metas
- Entorno -> analizar usuario y ambiente de trabajo
- Proyecto -> restricciones impuestas por los involucrados (Costo, calidad, tiempo)

- **Problemas de Comunicación**

- Deficiente comunicación entre usuarios y técnicos
- Muchas comunidades involucradas
- Estructura adecuada de la información

- **Problemas de volatilidad**

- Requisitos no permanecen estáticos
- Negocio cambia con el tiempo (metas, objetivos, etc)

# PROBLEMAS PARA IDENTIFICAR REQUISITOS

No siempre son obvios y tienen diversas fuentes

- No son fáciles de identificar con palabras (Métodos formales)
- Muchos tipos distintos, a distinto nivel de detalle
- El número puede llegar a ser inmanejable
- Muchos Interesados y partes responsables
- Pueden ser afectados por el tiempo y cambiar a lo largo del ciclo de vida de desarrollo
- Se interrelacionan con otros
- Es difícil manejar un vocabulario común con expertos del negocio.

Estos problemas impactan directamente sobre el costo

## ¿Por qué?



# TIPOS DE REQUISITOS/ I

Existen varios tipos de requisitos como lo son:

- **Requisitos de Usuarios:** Necesidades que los usuarios expresan verbalmente.
- **Requisitos del Sistema:** Son los componentes que el sistema debe tener para realizar determinadas tareas.
- **Requisitos Funcionales:** Servicios que el sistema debe proporcionar.
- **Requisitos no funcionales:** Restricciones que afectan al sistema.

# TIPOS DE REQUISITOS/2

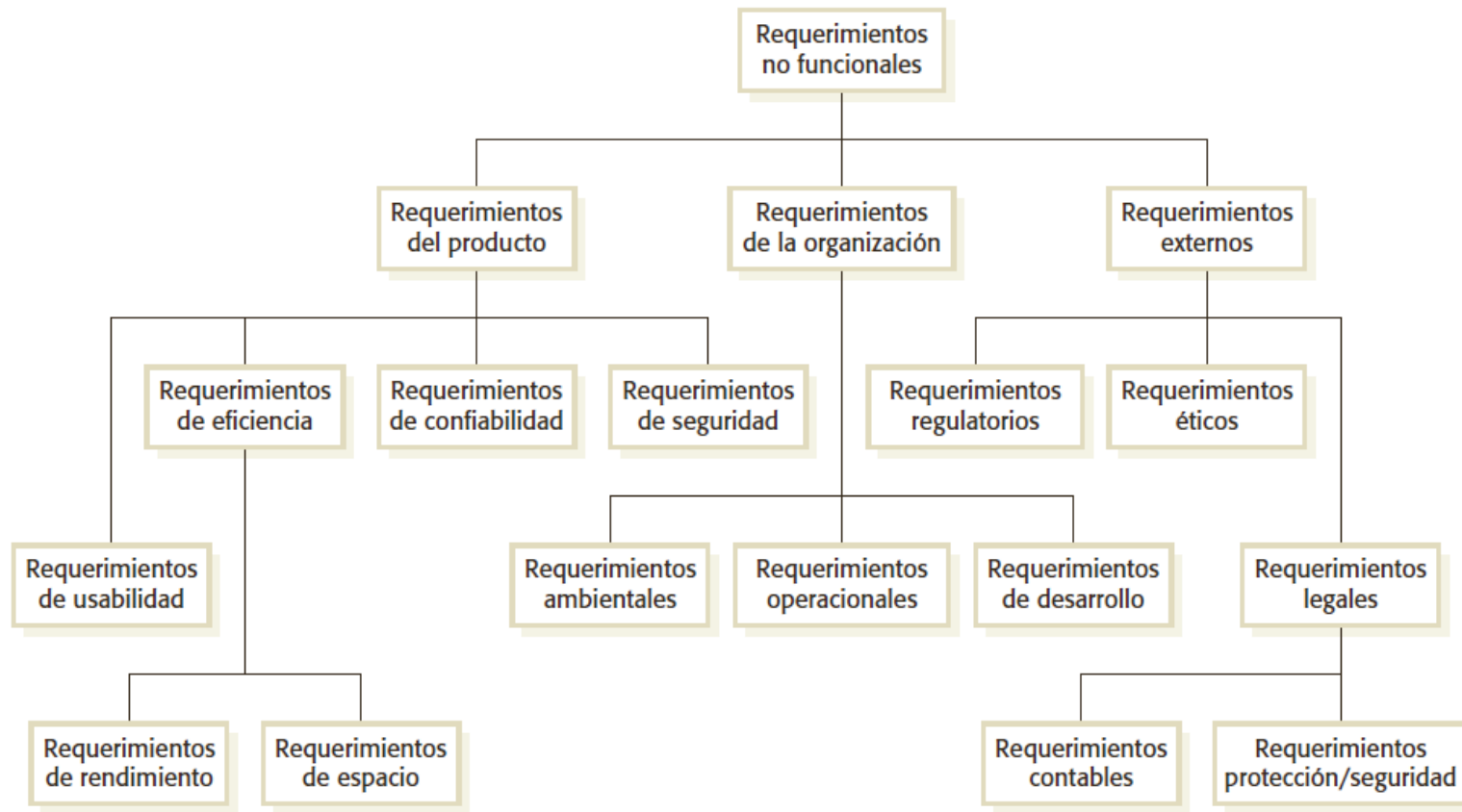
- **Funcionales:**
  - Especifica una acción que debe ser capaz de realizar el sistema, sin considerar restricciones físicas.
  - Describen la funcionalidad o los servicios que se espera proveerá el sistema.
- **No funcionales:**
  - Especifica restricciones físicas sobre un requisito funcional (rendimiento, plataforma, fiabilidad)
  - No se refieren directamente a las funciones específicas que entrega el sistema, sino a las propiedades emergentes de éste como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento.
  - A nivel hardware y Redes.

# REQUISITOS FUNCIONALES - EJEMPLO

## Sistema de venta de Música Online

- Los usuarios comprarán créditos para adquirir canciones.
- El sistema debe registrar la información de los usuarios y los créditos que poseen.
- Los usuarios buscarán las canciones que deseen y las pagarán con créditos.
- El sistema debe almacenar información sobre las canciones que se pueden adquirir y su precio en Créditos.

# REQUISITOS NO FUNCIONALES



## REQUISITOS NO FUNCIONALES – EJEMPLO

- El sistema debe visualizarse y funcionar correctamente en cualquier navegador, especialmente en Internet Explorer, Firebird, Mozilla y Nautilus.
- El sistema debe cumplir las disposiciones recogidas en la Ley Orgánica de Datos Personales y en el Reglamento de medidas de seguridad.
- El sistema no debe tardar más de cinco segundos en mostrar los resultados de una búsqueda. Si se supera este plazo, el sistema detiene la búsqueda y muestra los resultados encontrados.

# REQUISITOS NO FUNCIONALES

## Métricas para Identificar RNF

| PROPIEDAD               | MEDIDA   |
|-------------------------|--|
| <i>Rapidez</i>          | <ul style="list-style-type: none"><li>• Transacciones procesadas por segundo</li><li>• Tiempo de respuesta al usuario y a eventos</li><li>• Tiempo de actualización de la pantalla</li></ul>                           |
| <i>Tamaño</i>           | <ul style="list-style-type: none"><li>• KB's</li><li>• Tamaño de RAM</li></ul>   |
| <i>Facilidad de uso</i> | <ul style="list-style-type: none"><li>• Tiempo de capacitación</li><li>• Número de ventanas de ayuda</li></ul>   |
| <i>Fiabilidad</i>       | <ul style="list-style-type: none"><li>• Tiempo promedio entre fallas</li><li>• Probabilidad de no disponibilidad</li><li>• Tasa de ocurrencia de las fallas</li><li>• Disponibilidad</li></ul>                         |
| <i>Robustez</i>         | <ul style="list-style-type: none"><li>• Tiempo de reinicio después de fallas</li><li>• Porcentaje de eventos que provocan las fallas</li><li>• Probabilidad de corrupción de los datos después de las fallas</li></ul> |
| <i>Portabilidad</i>     | <ul style="list-style-type: none"><li>• Porcentaje de declaraciones dependientes del objetivo</li><li>• Número de sistemas objetivo</li></ul>  |

# REQUISITOS

## Posibles actores en la Ingeniería de Requisitos

- **Usuario Final**
- **Usuario Líder**
- **Personal de Mantenimiento**
- **Analistas y Programadores**
- **Personal de Pruebas**
- **Administradores de proyecto**
- **Diseñadores**

# GESTIÓN DE LOS REQUISITOS

- ❑ La Gestión de requisitos es el proceso de gestionar los cambios en los requisitos de un sistema y se integra en la Gestión del proyecto.
- ❑ Los requisitos de un sistema evolucionan, los sistemas no son estables.
- ❑ Para su gestión, hay que tener en cuenta algunos aspectos:
  - ❑ Requisitos estables y volátiles (mutables, emergentes, organizacionales y de compatibilidad)
  - ❑ Identificación y almacenamiento
  - ❑ Gestión del cambio
  - ❑ Trazabilidad
  - ❑ Gestión de riesgos



# BENEFICIOS DE LA GESTIÓN DE LOS REQUISITOS

- Mejor control de proyectos complejos.
- Mejora en la calidad del software y en la satisfacción del cliente.
- Reducción en los retrasos y en los costos del proyecto.
- Mejora en la comunicación del equipo.
- Facilita la conformidad con estándares y regulaciones.

# REQUISITOS DE CASOS DE USO

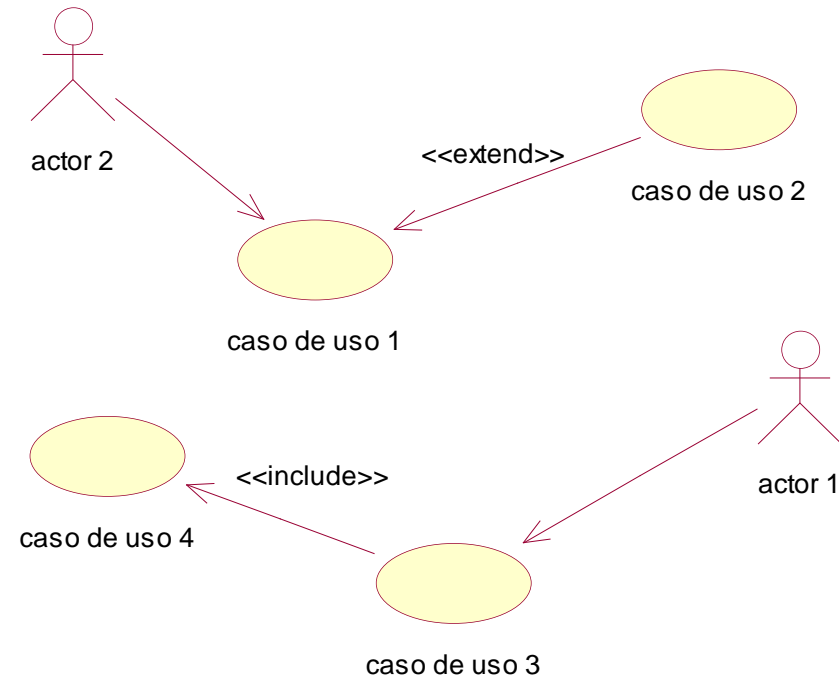
- Los casos de uso son una técnica para la especificación de requisitos funcionales propuesta inicialmente por Jacobson (1992).
- Los casos de uso presentan una ventaja sobre la descripción textual de los requisitos funcionales.
- **Los casos de uso son, ante todo, requisitos funcionales.**
- A pesar de ser una técnica ampliamente aceptada, existen múltiples propuestas para su realización.

# ¿QUÉ ES UN CASO DE USO?

- Los casos de uso son una técnica que se utiliza para documentar los requerimientos funcionales de un sistema desde el punto de vista de los usuarios.
- Los casos de uso responden a la pregunta: ¿Qué se supone que el sistema debe hacer para los usuarios?
- Un caso de uso es un texto muy simple con cierto formato que describe cómo se debería comportar un sistema ante la interacción con uno o más usuarios.

**Los casos de uso no son “orientados a objetos”**

# DIAGRAMA DE CASOS DE USO



El valor de los casos de uso está en su descripción detallada (no en el dibujo...)

# CASOS DE USO

Describe tanto lo que hace el actor como lo que hace el sistema cuando interactúa con él .

Están acotados al uso de una determinada funcionalidad, claramente diferenciada, del sistema.

Un caso de uso realiza cierto trabajo cuyo efecto es tangible para un actor.



# CASOS DE USO - ACTORES

Un actor es alguien o algo que interactúa con el sistema (una persona, una organización, un programa o sistema de hardware o software).

Un actor estimula al sistema con algún evento o recibe información del sistema.

Un actor es externo al sistema.

Un actor cumple un rol definido (Ej.: Cliente, Banco, empleado).

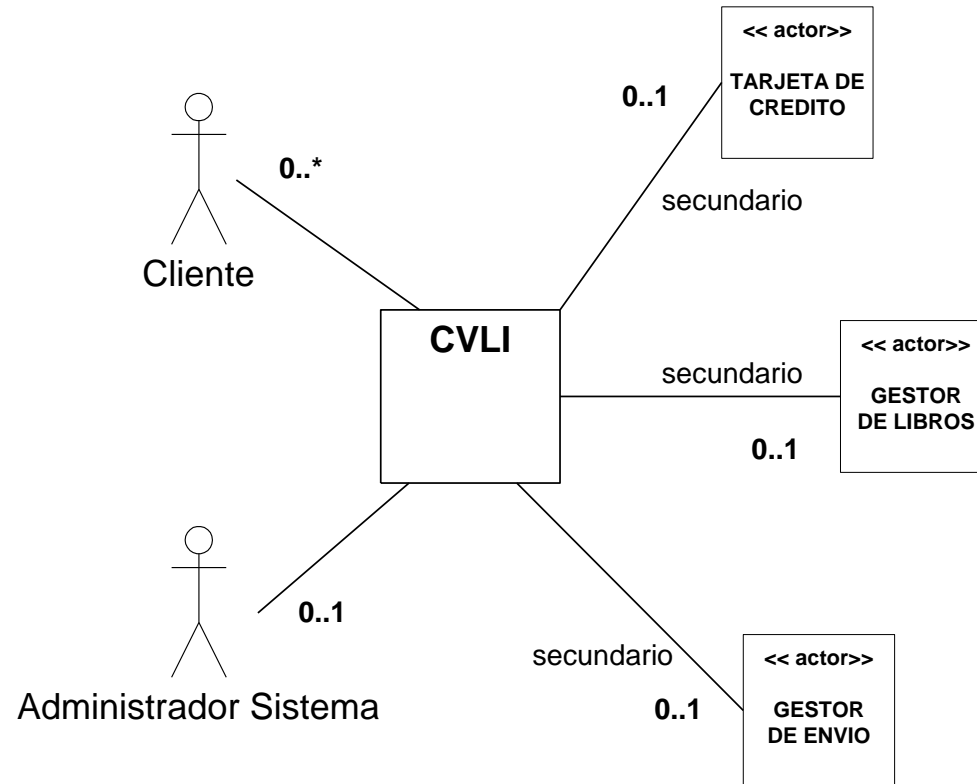
# TIPO DE ACTORES

**Actores primarios:** utilizan las funciones principales del sistema.

**Actores secundarios:** efectúan tareas administrativas o de mantenimiento.

# DIAGRAMA DE CONTEXTO

Permite determinar las fronteras del sistema





# ¿CÓMO ENCONTRAR ACTORES?

Una buena técnica para encontrar posibles actores es responder las siguientes preguntas:

- ¿Quién usa el sistema?
- ¿Quién instala o mantiene al sistema?
- ¿Qué otros sistemas utilizan al sistema?
- ¿Quién recibe información del sistema?
- ¿Quién le provee información al sistema?

# CASOS DE USO Y ESCENARIOS

Un caso de uso no trivial describe un conjunto de secuencias, no una única secuencia. Es conveniente separar el flujo principal de los flujos alternativos.

Por ejemplo: el caso de uso Contratar Empleado podría tener variantes: contratar externos (Escenario más frecuente), traslado dentro de la misma empresa, contratar extranjeros. Cada una de estas variantes se puede expresar en una secuencia diferente.

Cada secuencia específica del caso de uso se denomina **Escenario**.

Un **escenario** es una secuencia específica de acciones entre los actores y el sistema (es una instancia de un caso de uso).

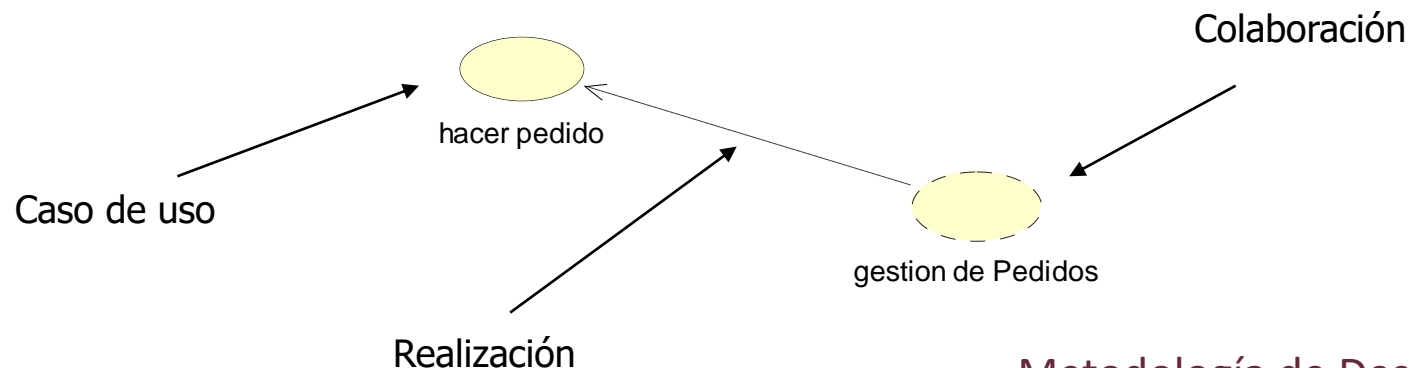
# CASOS DE USO Y COLABORACIONES

Un caso de uso captura el comportamiento esperado del sistema sin tener que especificar cómo se implementará.

Es importante separar el **análisis** (que especifica un comportamiento) de la **implementación** (que especifica cómo se lleva a cabo ese comportamiento).

De todas maneras, un caso de uso deberá implantarse y esto lo hará mediante una sociedad de clases (incluyendo su estructura estática y dinámica) que se modela como una **colaboración**.

Una realización especifica un contrato entre el caso de uso y su colaboración.



# PREY POST CONDICIONES

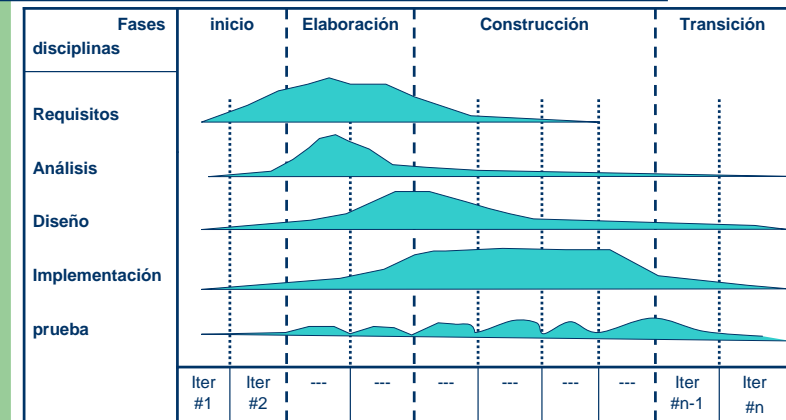
**Pre condiciones:** establece lo que siempre debe cumplirse antes de comenzar un caso de uso. No se prueban en el caso de uso, se asumen que son verdaderas.

**Post condiciones:** establece qué debe cumplirse cuando el caso de uso se completa con éxito.

# LOS CASOS DE USO Y LAS FASES EN EL UP

- En la fase de **inicio** se reconocen la mayoría de los casos de uso, pero no su descripción detallada.
- En la fase de **elaboración**, se refinan en las sucesivas iteraciones.
- En la fase de **construcción**, se escriben casos de uso menores.
- En la fase de **transición**, no se describen casos de uso.

## El Proceso Unificado Iterativo e incremental





Fin de la clase



**UAI**

**Universidad Abierta  
Interamericana**