



# Ingeniería de Software I – Clase 2 - 2024



```
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True
```

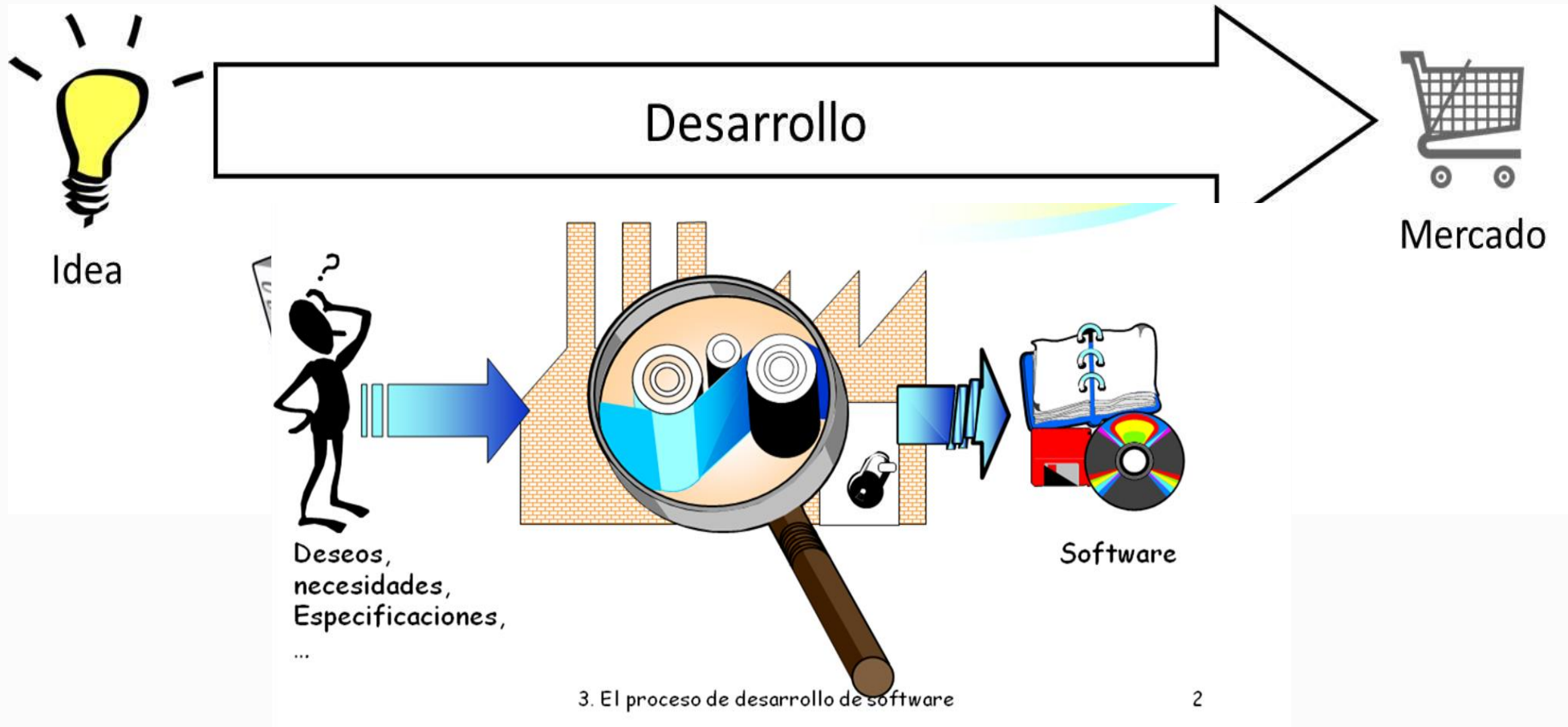
```
#selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.name))  
mirror_ob.select = 0  
= bpy.context.selected_objects  
data.objects[one.name].select  
print("please select exactly one object")
```

```
-- OPERATOR CLASSES ----
```

```
operator):
```

# Requerimientos de software

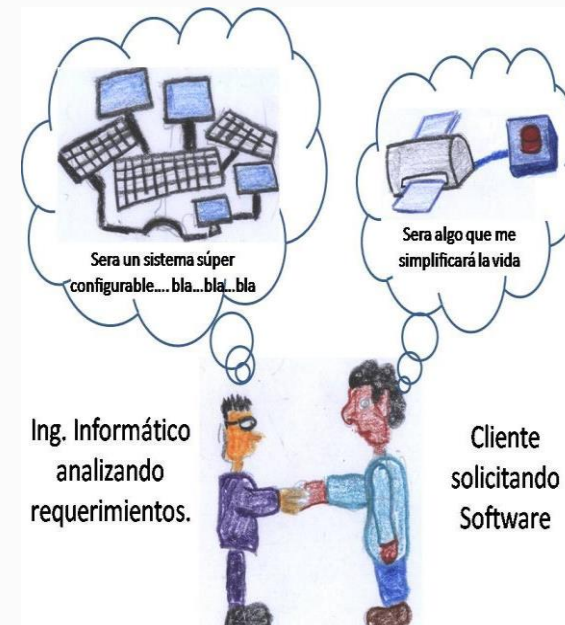
# ¿Qué es un proceso de software?



# Requerimientos

---

Un requerimiento (o requisito) es una característica del sistema o una descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer el propósito del sistema



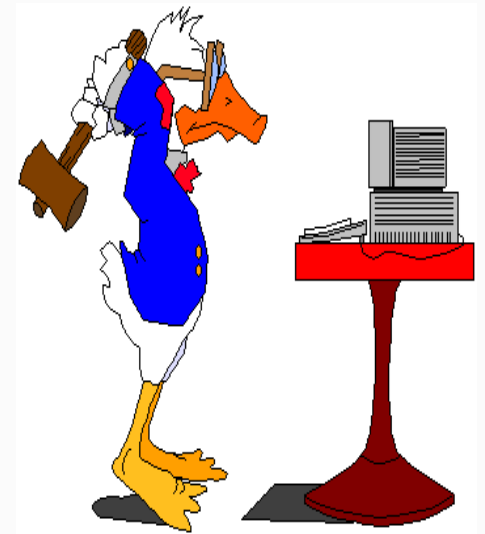


# Requerimientos

---

## Impacto de los errores en la etapa de requerimientos

- El software resultante puede no satisfacer a los usuarios
- Las interpretaciones múltiples de los requerimientos pueden causar desacuerdos entre clientes y desarrolladores
- Puede gastarse tiempo y dinero construyendo el sistema erróneo



# Requerimientos -Tipos

---

## Requerimientos funcionales

- Describen una interacción entre el sistema y su ambiente. Cómo debe comportarse el sistema ante determinado estímulo.
- Describen lo que el sistema debe hacer, o incluso cómo NO debe comportarse.
- Describen con detalle la funcionalidad del mismo.
- Son independientes de la implementación de la solución.
- Se pueden expresar de distintas formas

## Requerimientos no funcionales

- Describen una restricción sobre el sistema que limita nuestras elecciones en la construcción de una solución al problema.

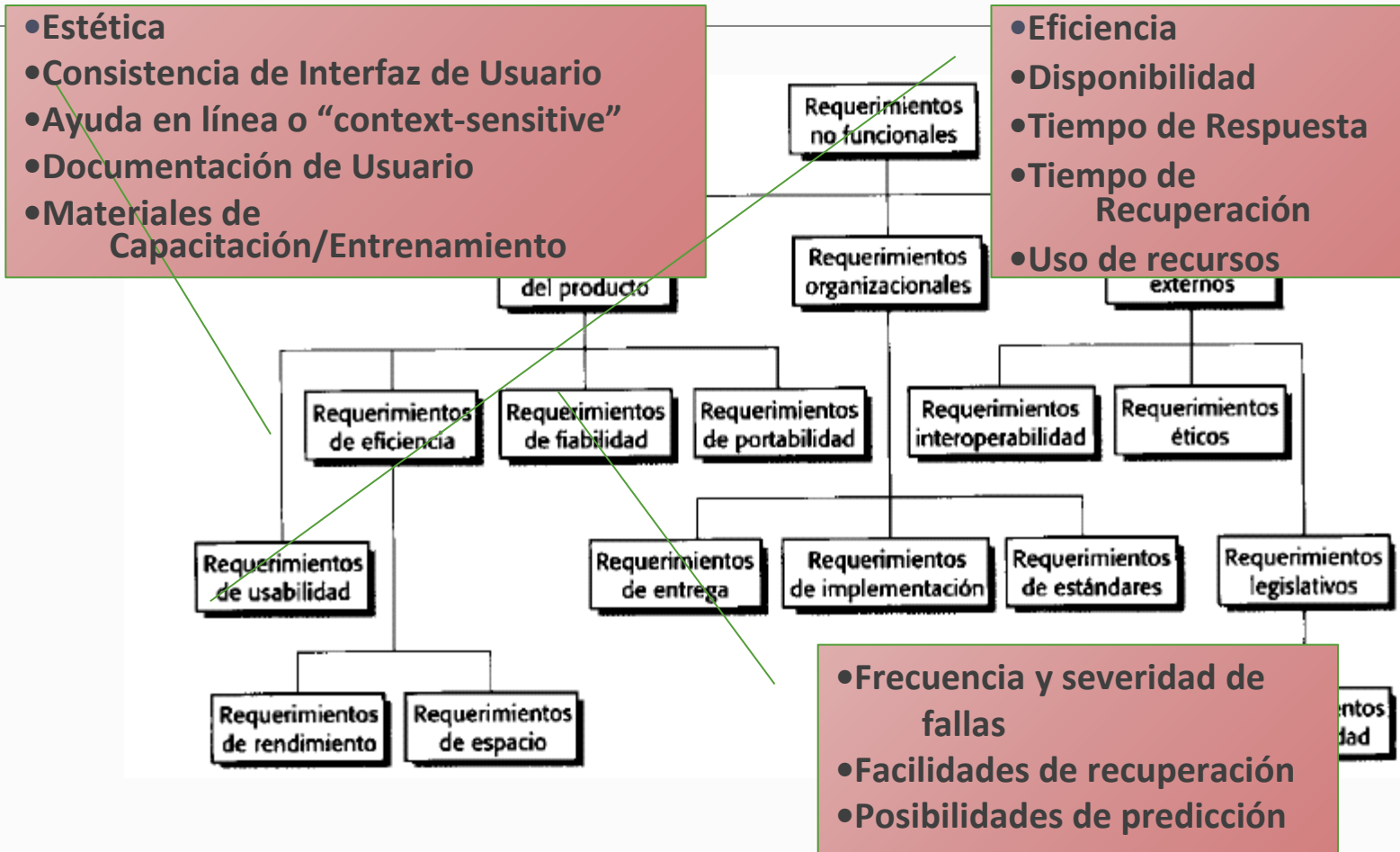
# Requerimientos - Tipos

---

## Requerimientos no funcionales

- Requerimientos del producto
  - Especifican el comportamiento del producto (usabilidad, eficiencia, rendimiento, espacio, fiabilidad, portabilidad).
- Requerimientos organizacionales
  - Se derivan de las políticas y procedimientos existentes en la organización del cliente y en la del desarrollador (entrega, implementación, estándares).
- Requerimientos externos
  - Interoperabilidad, legales, privacidad, seguridad, éticos.

# Requerimientos No Funcionales






# Ingeniería de Requerimientos



***Ingeniería de requerimientos*** es el proceso por el cual se transforman los requerimientos declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones”

La ingeniería de requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para el desarrollo de software.  La especificación es común entre todas las partes involucradas y en donde se describen las funciones que realizará el sistema.

# Ingeniería de Requerimientos

---

- También es el proceso mediante el cual se intercambian diferentes *puntos de vista para recopilar y modelar lo que el sistema va a realizar*. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos.”
- “Ingeniería de requerimientos” es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y *mantiene acuerdos* sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto”

# Ingeniería de Requerimientos

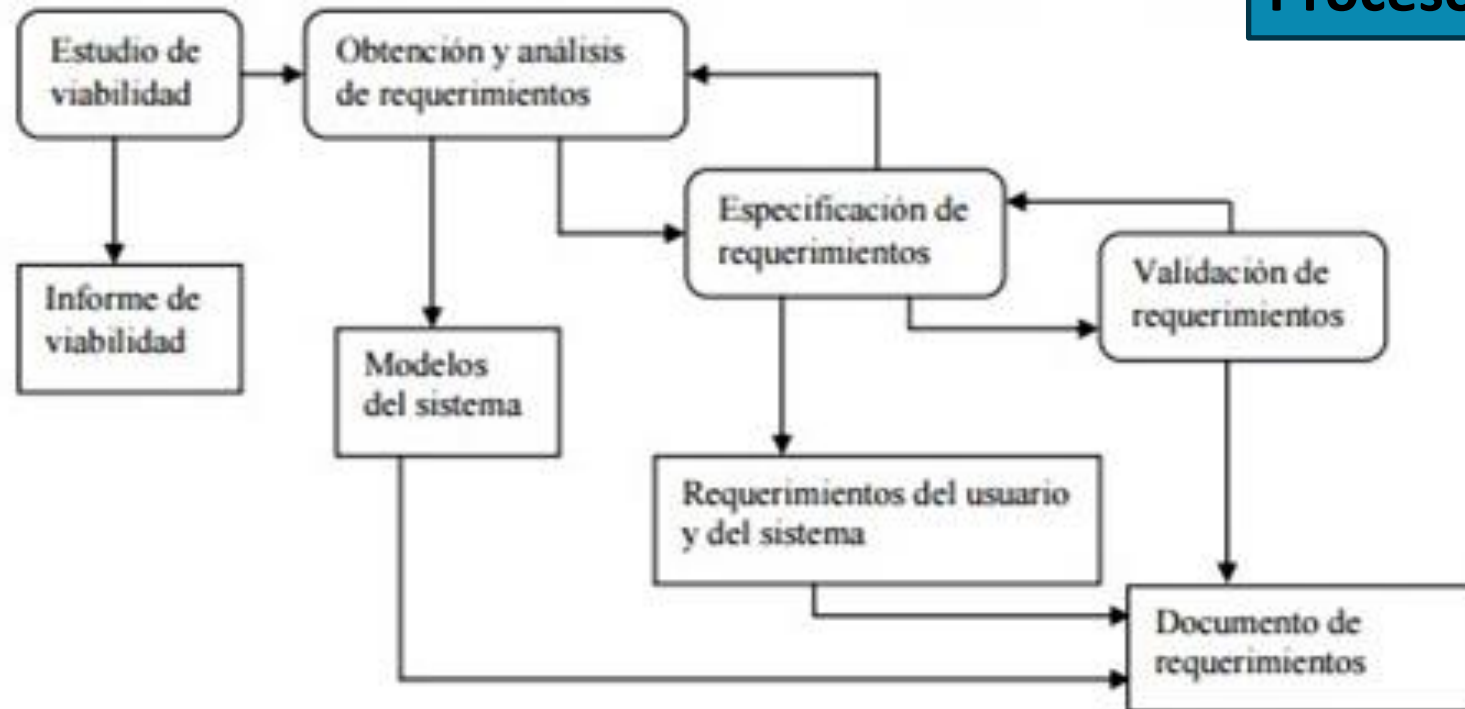
---

## Importancia

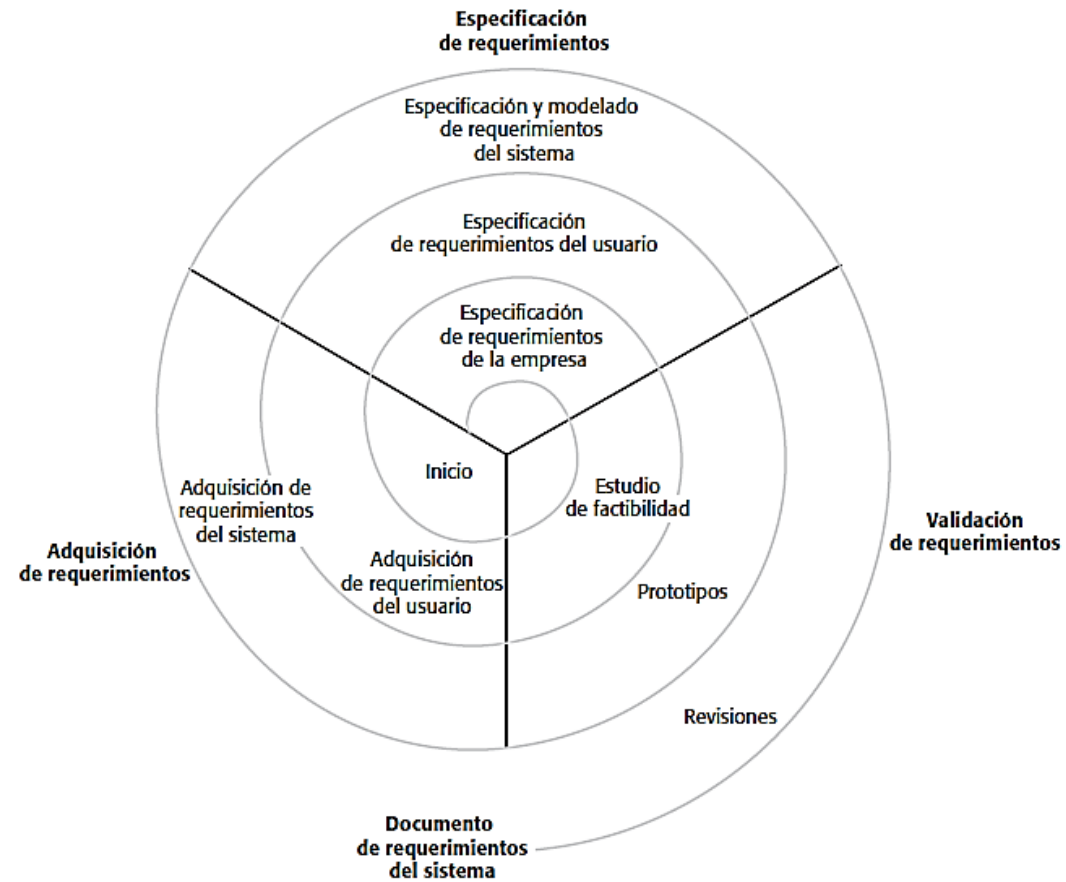
- Permite gestionar las necesidades del proyecto en forma estructurada
- Mejora la capacidad de predecir cronogramas de proyectos
- Disminuye los costos y retrasos del proyecto
- Mejora la calidad del software
- Mejora la comunicación entre equipos
- Evita rechazos de usuarios finales.

# Ingeniería de Requerimientos

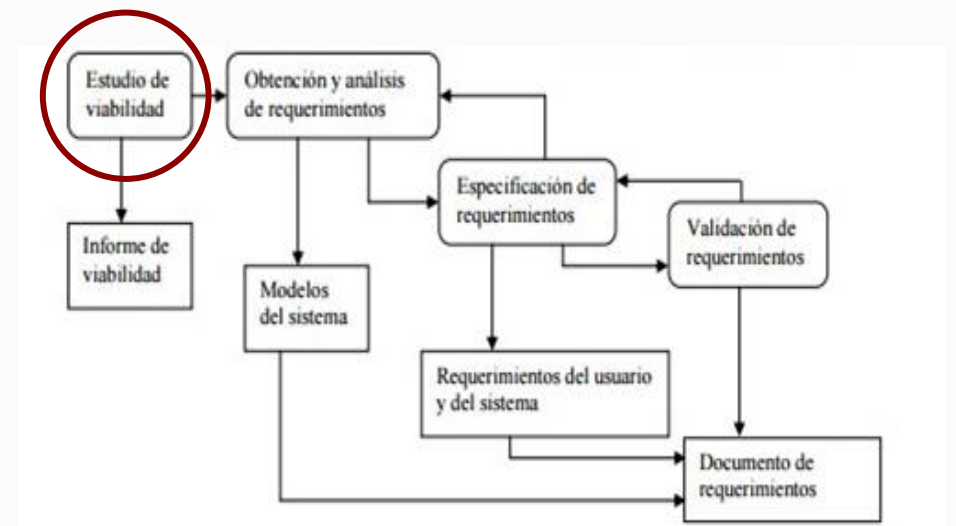
## Proceso secuencial



# Ingeniería de Requerimientos



**En la práctica , es un proceso iterativo donde las actividades se entrelazan**



# Estudio de Viabilidad



# Estudio de Viabilidad

---

Principalmente para sistemas nuevos

A partir de una descripción resumida del sistema se elabora un informe que recomienda la conveniencia o no de realizar el proceso de desarrollo

Responde a las siguientes preguntas:

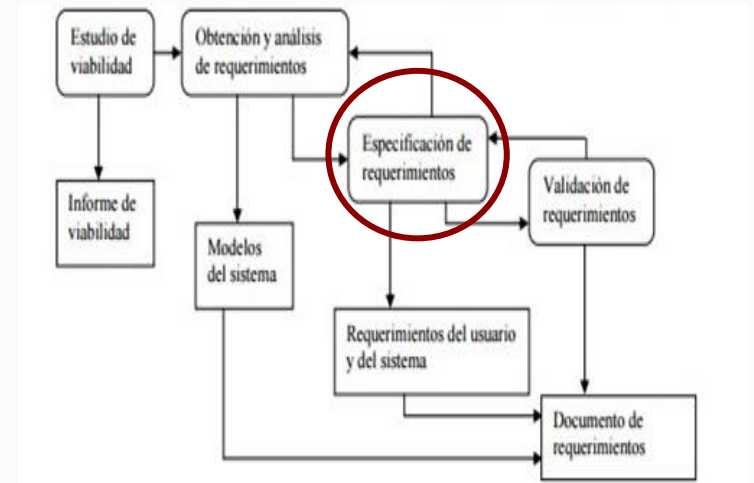
- ¿El sistema contribuye a los objetivos generales de la organización? (Si no contribuye, entonces no tiene un valor real en el negocio)
- ¿El sistema se puede implementar con la tecnología actual?
- ¿El sistema se puede implementar con las restricciones de costo y tiempo?
- ¿El sistema puede integrarse a otros que existen en la organización?

# Estudio de Viabilidad

---

- Una vez que se ha recopilado toda la información necesaria para contestar las preguntas anteriores se debería hablar con las fuentes de información para responder nuevas preguntas y luego se redacta el informe, donde debería hacerse una recomendación sobre si debe continuar o no el desarrollo.





# Especificación de requerimientos

# Especificación de Requerimientos

## Propiedades de los Requerimientos

- **Necesario:** Su omisión provoca una deficiencia.
- **Conciso:** Fácil de leer y entender
- **Completo:** No necesita ampliarse
- **Consistente:** No contradictorio con otro
- **No ambiguo:** Tiene una sola implementación
- **Verificable:** Puede testearse a través de inspecciones, pruebas, etc.



# Especificación de Requerimientos

---

## Objetivos

- Permitir que *los desarrolladores expliquen cómo han entendido lo que el cliente pretende del sistema*
- Indicar a *los diseñadores qué funcionalidad y características va a tener el sistema resultante*
- Indicar al *equipo de pruebas qué demostraciones llevar a cabo para convencer al cliente de que el sistema que se le entrega es lo que había pedido.*

# Especificación de Requerimientos

---

- Correcta
- No ambigua
- Completa
- Verificable
- Consistente
- Comprensible por los consumidores
- Modificable
- Rastreable
- Independiente del diseño
- Anotada
- Concisa
- Organizada
- Utilizable en operación y mantenimiento



# Especificación de Requerimientos

---

- Documento de definición de requerimientos  
Listado completo de todas las cosas que el cliente espera que haga el sistema propuesto
- Documento de especificación de requerimientos  
Definición en términos técnicos
- Documento de especificación de requerimientos de Software IEEE Std. 830-1998 (SRS)  
Objetivo:  
Brindar una colección de buenas prácticas para escribir especificaciones de requerimientos de software (SRS).  
Se describen los contenidos y las cualidades de una buena especificación de requerimientos.

# Especificación de Requerimientos

---

- Aspectos básicos de una especificación de requerimientos

## Funcionalidad

¿Qué debe hacer el software?

## Interfaces Externas

¿Cómo interactuará el software con el medio externo (gente, hardware, otro software)?

## Rendimiento

Velocidad, disponibilidad, tiempo de respuesta, etc.

## Atributos

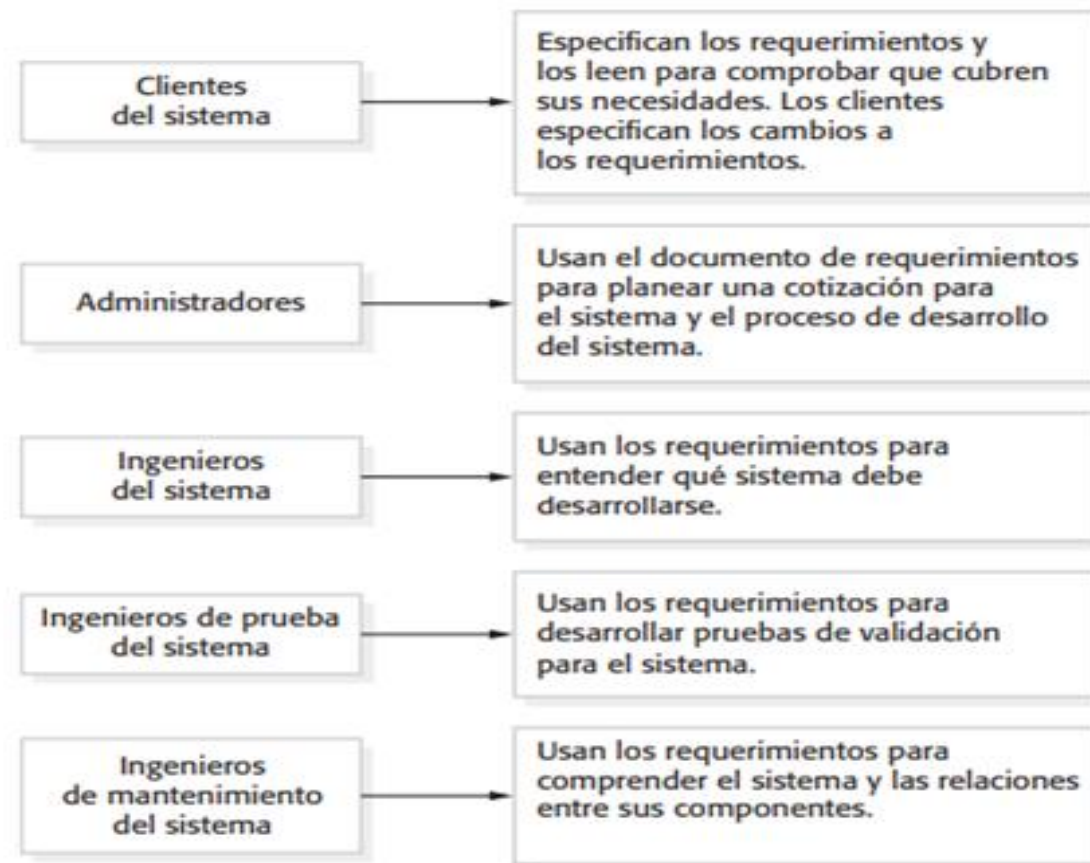
Portabilidad, seguridad, mantenibilidad, eficiencia

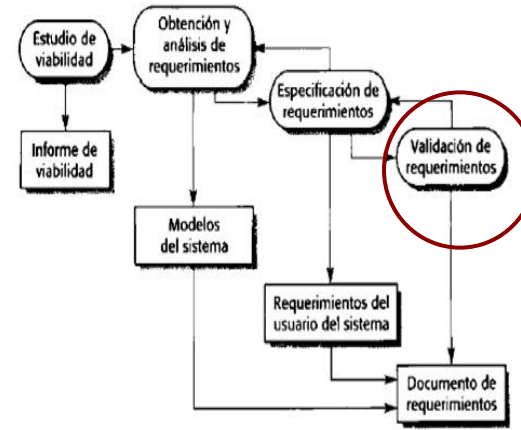
## Restricciones de Diseño

Estándares requeridos, lenguaje, límite de recursos, etc.

# Especificación de Requerimientos

- Usuarios de un documento de requerimientos





# Validación de requerimientos

# Validación de requerimientos

---

Es el proceso de *certificar la corrección del modelo de requerimientos contra las intenciones del usuario.*

Trata de *mostrar que los requerimientos definidos son los que estipula el sistema. Se describe el ambiente en el que debe operar el sistema.*

Es importante, *porque los errores en los requerimientos pueden conducir a grandes costos si se descubren más tarde*

# Validación de requerimientos

Definición de la IEEE

**Validación:** Al final del desarrollo evaluar el software para asegurar que el software cumple los requerimientos

**Verificación:** El software cumple los requerimientos correctamente

Sobre estas definiciones:

La validación sólo se puede hacer con la activa participación del usuario

Validación: hacer el software correcto

Verificación: hacer el software correctamente

26



# Validación de requerimientos

---

## *¿Es suficiente validar después del desarrollo del software?*

- La evidencia estadística dice que NO
- Cuanto más tarde se detecta, más cuesta corregir (Boehm)
- Bola de nieve de defectos
- Validar en la fase de especificación de requerimientos puede ayudar a evitar costosas correcciones después del desarrollo

## *¿Contra qué se verifican los requerimientos?*

- No existen “los requerimientos de los requerimientos”
- No puede probarse formalmente que un Modelo de Requerimientos es correcto. Puede alcanzarse una convicción de que la solución especificada en el modelo de requerimientos es el correcto para el usuario.

# Validación de requerimientos

---

## *Comprenden*

- Verificaciones de validez (para todos los usuarios)
- Verificaciones de consistencia (sin contradicciones)
- Verificaciones de completitud (todos los requerimientos)
- Verificaciones de realismo (se pueden implementar)
- Verificabilidad (se puede diseñar conjunto de pruebas)

# Técnicas de validación

---

Pueden ser manuales o automatizadas

- Revisiones de requerimientos (formales o informales)
  - Informales : Los desarrolladores deben tratar los requerimientos con tantos stakeholders como sea posible.
  - Formal : El equipo de desarrollo debe conducir al cliente, explicándole las implicaciones de cada requerimiento
- Antes de una revisión formal, es conveniente realizar una revisión informal.
- Construcción de prototipos
- Generación de casos de prueba



# Técnicas de Especificación de Requerimientos de software

# Técnicas de Especificación de Requerimientos

---

- Estáticas

Se describe el sistema a través de las *entidades u objetos, sus atributos y sus relaciones con otros. No describe cómo las relaciones cambian con el tiempo.*

Cuando el tiempo no es un factor mayor en la operación del sistema, es una descripción útil y adecuada.

Ejemplos: Referencia indirecta, Relaciones de recurrencia, Definición axiomática, Expresiones regulares, Abstracciones de datos, entre otras.

# Técnicas de Especificación de Requerimientos

---

- Dinámicas

Se considera un sistema en función de los cambios que ocurren a lo largo del tiempo.

Se considera que el sistema está en un estado particular hasta que un estímulo lo obliga a cambiar su estado.

Ejemplos: Tablas de decisión, Diagramas de transición de estados, Tablas de transición de estados, Diagramas de persianas, Diagramas de transición extendidos, Redes de Petri, entre otras.

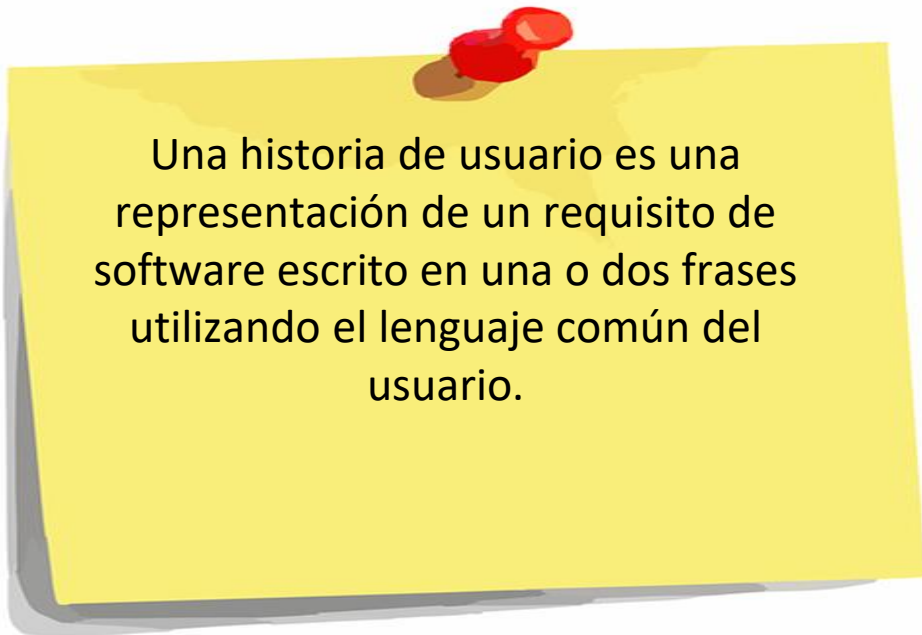


# Técnicas de Especificación de Requerimientos

## HISTORIAS DE USUARIO

# Historias de usuario - Definición

---



Una historia de usuario es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del usuario.

- Una historia de usuario es una descripción corta y simple de un requerimiento de un sistema, que se escribe en lenguaje común del usuario y desde su perspectiva.
- Son utilizadas en las metodologías de desarrollo ágiles (Ejemplo: XP, SCRUM) para la especificación de requerimientos  
Acompañadas de las discusiones con los usuarios y las pruebas de validación

# Historias de usuario – Conceptos

---



**Debe ser limitada**, ésta debería poder escribirse sobre una nota adhesiva pequeña.

Son **una forma rápida de administrar los requisitos** de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos.

Permiten **responder rápidamente a los requisitos cambiantes**.

Al momento de implementar las historias, los desarrolladores deben tener la posibilidad de **discutirlas con los clientes**.

Generalmente se espera que la **estimación de tiempo** de cada historia de usuario se sitúe entre unas **10 horas y un par de semanas**.

Estimaciones **mayores a dos semanas** son indicativo de que la historia es muy compleja y debe ser **dividida en varias historias**.

# Historias de usuario – Forma de redactarlas

---

Si bien el estilo puede ser libre, la historia de usuario debe responder a tres preguntas:

¿Quién se beneficia?

¿Qué se quiere?

¿Cuál es el beneficio?

Esquema:

Como (**rol**) quiero (**algo**) para (**beneficio**).

Ejemplos:

- Como **usuario registrado** quiero loguearme para **empezar a utilizar la aplicación**.
- Como **secretaria** quiero imprimir el listado de turnos asignados en una fecha determinada para **guardar la información de los mismos**.

# Historias de usuario – Forma de redactarlas

---

Como **Cliente** quiero **suscribirme por medio del sitio web** para **obtener un nuevo plan de T.V. por cable**

Como **Vendedor** quiero **registrar los productos y cantidades que me solicita un cliente** para **crear un pedido de venta**.

Como **Analista de compras** quiero **que el sistema notifique vía correo electrónico a los proveedores** para **avisar que se ha enviado una cotización de licitación**

# Historias de usuario – Características

---

**Independientes unas de otras:** De ser necesario, combinar las historias dependientes o buscar otra forma de dividir las historias de manera que resulten independientes.

**Negociables:** La historia en sí misma no es lo suficientemente explícita como para considerarse un contrato, la discusión con los usuarios debe permitir esclarecer su alcance y éste debe dejarse explícito bajo la forma de pruebas de validación.

**Valoradas por los clientes o usuarios:** Los intereses de los clientes y de los usuarios no siempre coinciden, pero en todo caso, cada historia debe ser importante para alguno de ellos más que para el desarrollador.

# Historias de usuario – Características

---

**Estimables:** Un resultado de la discusión de una historia de usuario es la estimación del tiempo que tomará completarla. Esto permite estimar el tiempo total del proyecto.

**Pequeñas:** Las historias muy largas son difíciles de estimar e imponen restricciones sobre la planificación de un desarrollo iterativo. Generalmente se recomienda la consolidación de historias muy cortas en una sola historia.

**Verificables:** Las historias de usuario cubren requerimientos funcionales, por lo que generalmente son verificables. Cuando sea posible, la verificación debe automatizarse, de manera que pueda ser verificada en cada entrega del proyecto.

# Historias de usuario - Criterios de aceptación

---

- Un criterio de aceptación es el criterio por el cual se define si una historia de usuario fue desarrollada según la expectativa del Product Manager/Owner (como representante de los criterios del cliente) y se si puede dar como hecha.
- Deben ser definidos durante la etapa inicial antes de la codificación, acompañan a la historia de usuario, porque complementan la historia de usuario y ayudan al equipo de desarrollo a entender mejor cómo se espera que el producto se comporte.



# Historias de usuario - Criterios de aceptación

---

- Los criterios de aceptación son utilizados para expresar el resultado de las conversaciones del cliente con el desarrollador. El cliente debería ser quien las escriba más que el desarrollador.
- Representan el inicio de la definición del cómo. No están diseñados para ser tan detallados como una especificación de diseño tradicional.
- Si una historia de usuario tiene más de 4 criterios de aceptación, debe evaluarse subdividir la historia.
- Puede añadirse un número de escenario para identificar al criterio, asociado a la historia de usuario en cuestión.

# Historias de usuario – Plantilla

Reverso

- **CRITERIOS DE ACEPTACIÓN:**
  - **Escenario 1:** título del criterio.  
Dado <un contexto inicial>,  
Cuando <ocurre un evento>,  
Entonces <garantiza uno o más resultados>
  - **Escenario 2:** título del criterio.  
Dado <un contexto inicial>,  
Cuando <ocurre un evento>,  
Entonces <garantiza uno o más resultados>
  - ....
  - **Escenario N:** título del criterio.  
Dado <un contexto inicial>,  
Cuando <ocurre un evento>,  
Entonces <garantiza uno o más resultados>

# Historias de Usuario

**Id:** Matricular persona

**Título:** Como *persona* quiero matricularme al instituto para poder hacer los cursos

**Reglas de Negocio:**

- Un DNI no puede estar registrado dos veces con diferentes matrículas
- El pago debe realizarse con tarjeta de crédito

# Historias de Usuario

Reverso

Criterios de Aceptación (Matricular persona):

## Escenario 1: Matriculación exitosa

**Dado** que el DNI 22.222.222 no se encuentra matriculado y las condiciones son las adecuadas para un pago exitoso.

**Cuando** la persona ingresa: "Juan Perez", DNI 22.222.222, dirección 7 #123 y presiona "Matricularse"

**Entonces** el sistema redirige al usuario al pago de matrícula con tarjeta de crédito, espera respuesta, matricula a la persona y genera su número de matrícula.

## Escenario 2: Matriculación fallida por matriculado existente

**Dado** que el DNI 12.123.123 se encuentra matriculado

**Cuando** la persona ingresa: "Ana Díaz", DNI 12.123.123 y dirección 51 #1321 y presiona "Matricularse"

**Entonces** el sistema informa que la persona ya se encuentra matriculada

## Escenario 3: Matriculación fallida por error en pago

**Dado** que el DNI 22.222.222 no se encuentra matriculado y las condiciones no son las adecuadas para un pago exitoso.

**Cuando** la persona ingresa: "Juan Perez", DNI 22.222.222, dirección 7 #123 y presiona "Matricularse"

**Entonces** el sistema redirige al usuario al pago de matrícula con tarjeta de crédito, espera respuesta e informa que el pago no ha sido correcto por lo que no se pudo matricular a la persona.

# Historias de usuario - Beneficios

---

- Al ser muy corta, ésta representa requisitos del modelo de negocio que pueden implementarse rápidamente (días o semanas).
- Necesitan poco mantenimiento.
- Mantienen una relación cercana con el cliente.
- Permite dividir los proyectos en pequeñas entregas.
- Permite estimar fácilmente el esfuerzo de desarrollo.
- Es ideal para proyectos con requisitos volátiles o no muy claros.



# Historias de usuario - Limitaciones

---

- Sin criterios de aceptación pueden quedar abiertas a distintas interpretaciones haciendo difícil utilizarlas como base para un contrato.
- Se requiere un contacto permanente con el cliente durante el proyecto lo cual puede ser difícil o costoso.
- Podría resultar difícil escalar a proyectos grandes.
- Requiere desarrolladores muy competentes.



# Historias de usuario - Épicas

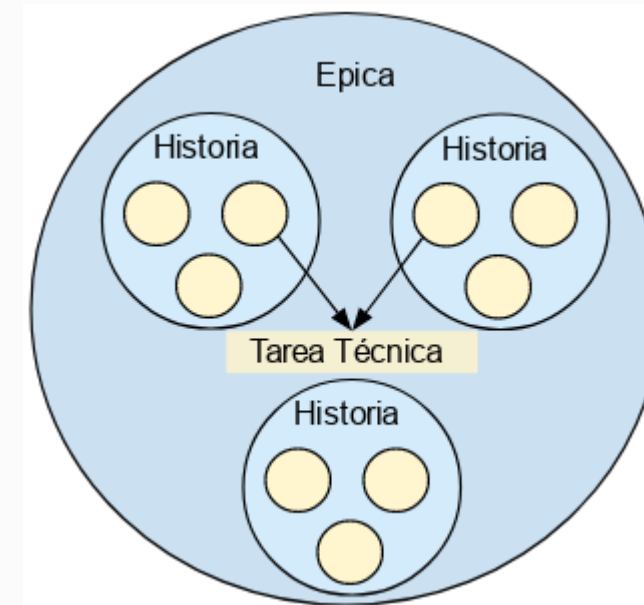
---

- Se denomina Épica a un conjunto de Historias de usuario que se agrupan por algún denominador común

La épica representa un objetivo alcanzable que nace de la necesidad del cliente.

Es un objetivo al que nos aproximamos y que esperamos alcanzar algún día.

La épica no es la funcionalidad.





# Ejemplo de Épicas

---

Épica:

*Como Vicepresidente de mercadeo y ventas quiero revisar el desempeño histórico de las ventas para identificar las regiones geográficas y productos de mejor desempeño*

Esta épica se puede subdividir en:

*Como VP de Mercadeo quiero seleccionar el período de tiempo en el cual realizaré la revisión de las ventas para analizar el desempeño*

*Como VP de Mercadeo quiero clasificar la información de ventas por región geográfica y productos para obtener la mejor zona de ventas*

Este ejemplo fue obtenido de Mountain Goat Software. Two Examples of Splitting Epics



# Ejemplo de Épicas

---

Épica:

*Como un operador hotelero quiero establecer las tarifas óptimas para las habitaciones de mi hotel.*

Esta épica contiene las siguientes HU:

Como un operador hotelero quiero establecer la tarifa óptima para cotizar las habitaciones en base a los precios del año anterior.

Como un operador hotelero quiero establecer la tarifa óptima para cotizar las habitaciones en base a las tarifas de otros hoteles comparables con el mío.

Este ejemplo fue obtenido de **Mountain Goat Software. Two Examples of Splitting Epics**

49

# Características de la épicas

---

- » Las épicas suelen abarcar varios equipos de desarrollo
- » Recogen normalmente muchas historias de usuario
- » Los clientes determinan si eliminan o añaden historias dentro de cada épica
- » Una épica sirve para estructurar los temas e iniciativas (objetivos)
- » Las épicas también sirven para dar flexibilidad y agilidad al proyecto

50

# Bibliografía

---

- Libros consultados para técnicas de especificación de requerimientos

Pfleeger, Capítulo 4 , Ingeniería de Software, Pearson Prentice Hall 2002

Sommerville Ian, Capítulo 4, Ingeniería de software, Addison Wesley 2011

Whitten y Bentley, Análisis de Sistemas Diseño y Métodos, Capítulo 6, Mc Graw Hill 2008.

**Mike Cohn, User Stories Applied, Addison Wesley 2004.**