

Requerimientos → Lo q' el program debe poder hacer

- Funcionales → interacción e/sistema y s. ambiente
cómo debe comportarse/NO comportarse
independiente de la implementación
- No Funcionales → Restricciones

- 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.

Ingeniería de requerimientos.

- Transformar requerimientos a especificaciones precisas y completas.
- Funciones, interfaces, rendimiento, limitaciones.
- Enfoque sistémico.
- Se intercambia por, se modela lo q' se va a hacer.

Viabilidad → A partir de la data q' tenemos del sistema, ver si conviene/se debe hacer

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?

Si no en concen, hay q' cambiarlo

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.

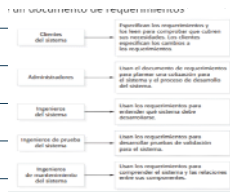


Permite q' developer explique lo q' entendieron
incluso a diseñadores lo q' va a ser el sistema

Especificación → define

- Para q' está el sistema? (funcionalidad)
- Interfaces externas
- Rendimiento (tiempo de respuesta, disponibilidad)
- Diseño

SRs → Documento de requerimientos



Validación de requerimientos → validación: ver q' software cumple requerimientos

→ Verificación: q' cumple los requerimientos bien

→ ¿Software validar después del desarrollo?

→ No, mejor antes lo después mejor

* Validación manual/automática

* Formal/informal?

* Protocolos | Casos de prueba

→ Bien visuales

Técnicas estáticas de especific. de requerimientos

→ Se describe sis. con estados/objetos, atributos y relaciones. No describe cómo van cambiando

Dinámicas

→ Qué pasa en el sistema cuando y lo todo

Técnicas de especificación de requerimientos

* Historias de usuario → Una sola cosa, bien concreta

→ Crear historias de usuario a partir de los requerimientos

→ Especificación se reduce a una frase

→ Metodologías de desarrollo ágiles

→ Debe ser verificable/corrob.

→ Forma rápida de administrar requisitos

→ Es discutir con cliente.

→ Historia de usuario tiene q' poder implementarse entre 10 h y una semana, → si es más largo, hay q' dividir

→ Al menos una historia (1 sola cosa)

es + historias

Para escribir historia

→ Qué se beneficia?

→ Qué se requiere

Criterio de aceptación.

¿Qué requiere

¿Cuál es el beneficio

[Como (rol) quiero (algo) para (beneficio)]

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

• Independientes uno de otros: si se alejan, juntar las dependencias o buscar otra forma de separarlas

• Negociables: historiano el explícito, Charlar con User.

• Valorable x cliente o user: interés no siempre coincide, historio de la info para ellos + que a desarrollador.

- 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.

¿Cómo lo tengo q' implementar

Planned → SD

→ Historia de user

→ Reglas de negocio ~ condiciones del modo de operación

lado B

CRITERIOS DE ACEPTACIÓN:

Escenario 1: título del criterio.
Dado <un contexto inicial>,
Cuando <ocurre un evento>,
Entonces <garantiza uno o mas resultados>

Escenario 2: título del criterio.
Dado <un contexto inicial>,
Cuando <ocurre un evento>,
Entonces <garantiza uno o mas resultados>

....
Escenario N: título del criterio.
Dado <un contexto inicial>,
Cuando <ocurre un evento>,
Entonces <garantiza uno o mas resultados>

lado B

- 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.



Épica → conjunto de historias de usuario q' se agrupan + denominan 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.