

# Para el primer irat

## Requisito

En ingeniería de desarrollo de sistemas un requisito es una necesidad documentada sobre el contenido forma o servicio de un determinado producto.

Estos requisitos se utilizan como datos en la parte parte del diseño del producto. Con esto podemos establecer el **que** pero no el **como** todavía.

Una cosa importante es que fucking significa elicitation: es una manera en que los humanos podemos darle información de forma fluida a otro ser humano

En informática es cuando podemos pasar de información sin dificultad a aquellos programas que se trabajan juntos.

## ¿Qué es un requisito?

- Condición o capacidad que se necesita para resolver un problema o lograr un objetivo. Por ejemplo si yo quiero aprobar la materia de eda tengo que tener la capacidad de aprobar los parciales.
- Condición o capacidad que esta formalmente documentada y expuesta
- Una condición o capacidad que debe ser conformada por el sistema
- Algo que el sistema debe hacer (estudiar) o debe poseer (los conocimientos y/o habilidad de java)

## Tipos de requisitos:

- **Requisito funcional:** algo que el sistema debe realizar
- **Requisito no funcional:** algo que el sistema debe cumplir por ejemplo tiene que ser rápido, útil, fácil de utilizar.

- Otros tipos de limitaciones externas por ejemplo el sistema operativo, pero que estas afectan al sistema en si. Por ejemplo también las leyes.

## Características que deben cumplir los requisitos:

- **No puede ser ambiguo:** tiene que ser claro y el texto no puede tener más de una interpretación
- **Conciso:** debe ser lo más claro posible en la parte técnica y especializado.
- **Consistente:** ningún requisito debe entrar en conflicto con otro requisito diferente.
- **Completo:** los requisitos deben contener toda la información necesaria.
- **Alcanzable:** Debe ser realista, que pueda ser alcanzado con el dinero, el tiempo y los recursos disponibles.
- **Verificable:** podemos decir si el requisito fue satisfecho o no.

La etapa en que se estudian los requisitos se llama Análisis de Requisitos.

## Metodología

En sistemas se ha planteado encontrar una metodología con el fin de mejorar la productividad en el desarrollo y aparte mejorar el producto.

En Ingeniería de software tenemos que realizar muchas tareas que están agrupadas a estas tareas se le denomina ciclo de vida, los cuales las etapas de este ciclo de vida son las siguientes:

- **Obtención de los requisitos:** lo primero es saber que motiva la creación de un nuevo software o la modificación de uno existente. Para esto tenemos que saber los recursos que tenemos tanto materiales como humanos. Se tiene que tener la información fuera del software tipo los usuarios finales que otros sistemas o dispositivos externos se vana utilizar (ejemplo un servicio de mensajería), que tipo de datos salen del sistema, y que datos si se juntan. Hay que ver también cuales son los puntos críticos, que es lo que complica el sistema lo que genere insatisfacción y las cosas que deben ser cumplidas en su

totalidad. Que funciones realizará el software y qué va a hacer en situaciones inesperadas.

- **Análisis de requisitos:** básicamente hay un cliente que explica que es lo que debería hacer el software o que problema pretende solucionar. El desarrollador debe realizar consultas. Los requisitos se deben tomar en cuenta pensando en el usuario final. El resultado del análisis de requisitos con el cliente se plasma en el documento ERS (especificación de requisitos del sistema) que este documento puede estar definida por varios estándares como CMMI. Este tiene un diagrama de entidad relación. La finalidad del análisis de requisitos: es brindar al usuario todo lo necesario para que pueda utilizar el software obteniendo buenos resultados. Tener un control en la etapa de creación del software en cuanto a tiempo de desarrollo y costos. Métodos más eficientes que permitan el aprovechamiento del software. Aumentar la calidad de software desarrollado al disminuir los riesgos de mal funcionamiento.
- Limitaciones: Un software no puede hacer de todo tiene predefinidas las funciones que puede realizar, más allá de eso no puede hacer.
- Especificación: Esta parte describe el comportamiento que va a tener el software una vez desarrollado. En este caso las técnicas son: casos de uso, historias de usuarios.
- Arquitectura: La integración de la infraestructura, desarrollo de aplicaciones, bases de datos y herramientas gerenciales, requieren capacidad y liderazgo. En esto entra el arquitecto. En este punto hay que poner en claro los aspectos lógicos y físicos de las salidas, modelos de organización y representación de datos, entradas y procesos que componen al sistema. El diseño arquitectónico debe permitir que el diseño sea validado en este caso por medio de diagramas de secuencias.

Alguno de los ejemplos de documentación utilizada es;

- Diagrama de clases
- De base de datos,
- de despliegue
- secuencia.

Las herramientas para el diseño y modelado de software se denominan CASE.

- Programación: Esta etapa no es necesariamente la que demanda más trabajo siempre depende a los lenguajes de programación utilizados así como el diseño

que se uso previamente,

- Desarrollo de la aplicacion: Para esto hay que conciderar cinco fases para tener una aplicacion o programa eficiente:
  - Desarrollo de la infraestructura: se desarrolla y organizan los elementos que fomraran parte de la infraestructura
  - Adaptacion del paquete: en esta etapa el objetivo es entende rel funcionamiento del paquete para poder utilizarlo en su maximo rendimiento. Todos los elementos del paquete son inspeccionados de manera detallada para evitar errores
  - Desarrollo de unidades de diseño de interactivas: En esta fase se realizan procedimientos que se ejecutan por un dialogo usuario siste,a.
    - Establecer las acciones que debe efectiar la unidad de diseño
    - La creacion de componentes para sus procedimientos
    - Ejecutar pruebas unitarias de integracion en la unidad de diseño
  - Desarrollo de unidades de diseño batch: en esta fase se utilizan los diagramas de flujo, estructura, decisiones etc. para que el programador tenga mayor comprension a la hora de programar.
  - desarrollo de unidades de diseño manuales: proyectar los procedimientos administrativos que se desarrollaran en torno a los componentes computalizados.
- Pruebas de software Consiste en que el programa cumpla con las tareas que fueron descriptos en la especificacion del problema. Una tecnica es probar por separado cada modulo de software (prueba unitaria) y luego probarlo de manera integral (prueba de integracion)
- Implementacion la implementacion es el proceso en el cual una especificación del sistema se convierte en ejecutable
- Documentacion Es toda la documentación para wue sea más fácil el mantenimiento del mismo
- Mantenimiento, fase dedicada al mantenimiento y mejoramiento del software lleva más tiempo que el desarrollo del software inicial. Una pequena parte consiste eneliminar errores; siendo la sigueinte parte el incoproarle nuevas funcionalidades y hacer frente a la evolución.

## Caso de uso:

Es la descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso. El diagrama de caso de uso representa a un sistema o subsistema como un conjunto de interacciones que se desarrollaran entre casos de uso y entre estos y sus actores, en respuesta a un evento que inicia el

## Definiciones básicas:

**Actores:** Se le llama actor a toda entidad externa al sistema que guarda una relacion con este y que le demanda una funcionalidad. Esto incluye a los humanos pero tambien incluye a todos los sistemas externos ademas de entidades abstractas como el tiempo.

## Tipo de relaciones:

- **Comunica:** Relacion entre un actor y un caso de uso que denota la participacion del actor en dicho caso de uso.
- **Usa:** relacion de dependencia entre dos casos de uso que denota la inclusion del comportamiento de un escenario en otro.
- **Generalizacion:** Indica que un caso de uso es una variante de otro.
- **Extiende:** Es un estereotipo de dependencia. Ofrece una forma de extensión mas controlada que la relacion de generalizacion.

## Un caso de uso debe:

- Describir una tarea de negocio que sirva a una meta de negocio
- Tener un nivel apropiado del detalle

- Ser bastante sencillo como para que un desarrollador lo elabore en un único lanzamiento.

## Limitaciones:

Los casos de uso pueden ser útiles para establecer requisitos de comportamiento pero no establecen los requisitos funcionales ni permiten determinar los que no lo son.

## Requisito Funcional:

Define un conjunto de entradas, comportamientos y salidas del sistema del software o sus componentes. Estos pueden ser calculos, detalles tecnicos, manipulacion de datos y otras funcionalidades especificas que se supone que un sistema debe cumplir.



Los requisitos funcionales establecen los comportamientos del software.

Conmumente un analista de requisitos genera requisitos funcionales despues de realizar los casos de uso.

Un requisito funcional tipico contiene un nombre un numero de serie unico y un resumen.

## Requisito no funcional o atributo de calidad:

especifica criterios que pueden usarse para juzgar la operacion de un sistema. son todos los requisitos que no describen informacion a guardar, ni funciones a realizar, sino características de funcionamiento. Por ejemplo: el tiempo de entrega del programa, el lenguaje o la cantidad de usuarios.

|:

1. Requisitos de calidad de ejecución, que incluyen seguridad usabilidad y otros medibles en tiempo de ejecución.
2. Requisitos de calidad de evolución como testeabilidad, extendibilidad o escalabilidad que evalúan en los elementos estáticos del sistema de software.

Algunos ejemplos no funcionales típicos:

- Rendimiento
- disponibilidad
- durabilidad
- Integridad de datos
- Documentación
- Privacidad

Historias de usuarios:

Es una explicación general informal escrita desde la perspectiva del usuario final.

La idea de esto es poner al usuario final en el centro de la conversación.

Hay que notar que los clientes no tienen que ser al final los usuarios finales.