***TÉCNICAS DE OBTENCIÓN DE REQUISITOS DEL SOFTWARE***

***¿Qué es un requisito?***

-Una condición o capacidad que debe cumplir o poseer un componente del sistema

-Una condición o capacidad que necesita un usuario para resolver un problema o alcanzar objetivos

-Una imagen, fórmula, modelo…

***Descripción de las propiedades de un sistema, de sus atributos y de cómo se debe comportar.***

-Restringen el proceso de desarrollo de un sistema.

-Un requisito debe describir:

las facilidades a nivel de usuario

las propiedades generales del sistema

las restricciones específicas

las restricciones de desarrollo

-Contienen:

problemas de la información, descripciones del comportamiento, propiedades y atributos, y restricciones.

***¿Qué describe un requisito?***

Una utilidad para el usuario

 “El tratamiento de textos ha de incluir la comprobación y corrección gramatical”

Una propiedad general del sistema

“El sistema ha de garantizar que la información personal solamente será accesible mediante autorización explícita”

 Una restricción general del sistema

 “El sensor ha de muestrearse 10 veces por segundo”

 Cómo llevar a cabo cierto cálculo

“Calificación final = nota examen + 2\*nota trabajo + 2/3 nota ejercicios”

Una restricción sobre el desarrollo del sistema

“El sistema ha de implementarse en C#”

***¿Por qué Ingeniería de Requisitos?***

-Base para el proceso de desarrollo del producto (sin planos no se hace una casa)

-Debe ser correcta y completa para construir el producto software adecuado

-Dificultades

-Los errores en los requisitos son costosos

-El enfoque = Qué y Cómo, Problema y Solución

Actividades para descubrir, documentar y mantener el conjunto de requisitos. Implica uso de técnicas sistemáticas e iterativas, asegurando que los requisitos del sistema estén bien.

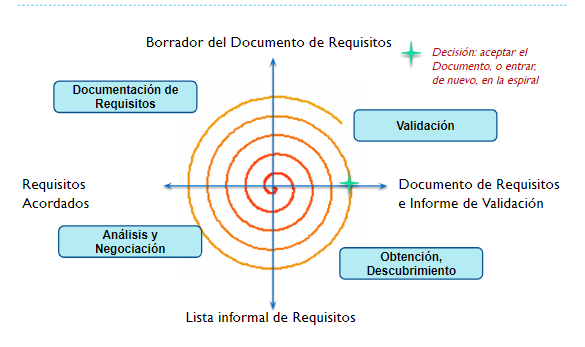
Un buen proceso de ingeniería de requisitos debe incluir:

Obtención de Requisitos.

consultas con los afectados por el sistema real

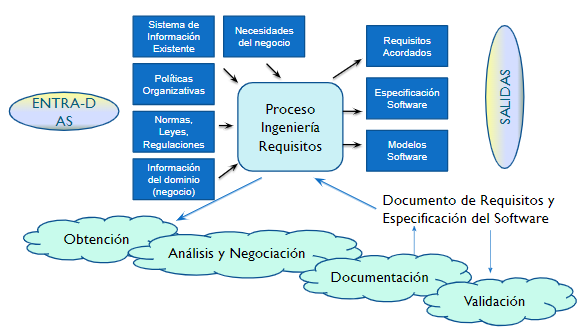
por los documentos que se manejan

por el conocimiento del dominio del sistema, y

estudios de mercado.

Análisis y negociación de requisitos.

Validación de requisitos.



***¿Participantes? Stakeholders***

-Un proyecto para hacer un producto software que ayuda a personas hacer su trabajo

-Para hacer el producto adecuado es necesario entender el trabajo, la gente que hace o influye y la meta

-Las stakeholders son personas que afectan o se ven afectadas por el producto de software y se benefician de su existencia.

-Análisis de las partes interesadas, son personas que tienen interés en el sistema o se ven afectadas por el entonces deben ser consultadas durante la obtención de requisitos. Estas partes incluyen grupos e individuos internos y externos a la organización. El cliente suele ser la parte interesada más aparente del sistema. En algunos casos, los usuarios reales del sistema pueden ser los más importantes. Si se ven afectadas los responsables de los estándares de procesos de trabajo, los clientes y los socios, también deben considerarse. También suele incluir la identificación de representantes clave de los usuarios y campeones del producto.

***Stakeholders - Identificarlos***

-Cualquiera que se vea afectada y beneficiada del producto o está interesada.

-Tres grandes grupos:

1. Clientes, también se les conoce como Patrocinadores, pagan el desarrollo Desarrollo a medida o producto a la venta.
2. Clientes compradores, compran el producto, si existen muchos clientes potenciales participa un representante
3. Usuario, utiliza el producto a medida trabajando para el cliente o si hay un producto a la venta el usuario es lo mismo que el cliente comprador.

Dentro de estos hay tres categorías:  
1. Los adquirentes del producto de software

2. Los proveedores de producto de software

3. Otras partes interesadas → Clientes que solicitan, compran y/o pagan por el producto.

→ Grupos de usuarios o usuarios finales, ellos usan el producto directamente o indirectamente

***Stakeholders - Mas sobre Usuarios***

-Atributos que el producto debe satisfacer:

Personas con discapacidad

No lectores o gente que habla otro idioma o que usan gafas

Personas que no suelen usar ordenador

Personas que transportan grandes paquetes o bebes

-Considerar las siguientes posibilidades:  
 ¿Cuánta ayuda necesita?

Experiencia tecnológica y habilidades intelectuales

Actitud hacia su trabajo, educación y habilidades lingüísticas

***Otros Stakeholders***

-Consultores externos o internos

-Gestores

-Equipo base

-Expertos en el Dominio

-Personal de Marketing

-Abogados

-Los proveedores incluyen individuos y equipos que forman parte de la organización que desarrolla el producto o son parte de las organizaciones que distribuyen el producto de software o están involucrados en otros métodos de entrega de productos (por ejemplo, subcontratación).

-El analista de requisitos es responsable de obtener los requisitos de los clientes, usuarios y otras partes interesadas, analizar los requisitos, escribir la especificación de los requisitos y comunicar los requisitos al desarrollo y otros.

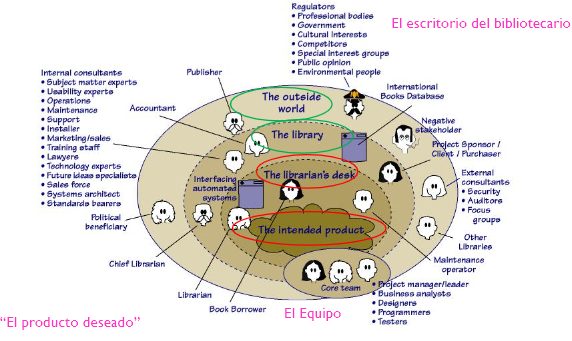
-Los diseñadores son responsables de traducir los requisitos en los diseños arquitectónicos y detallados del software. También implementar los diseños al crear los productos de software, si forma parte de un sistema grande, los diseñadores y desarrolladores de hardware también pueden estar interesados

-Los evaluadores utilizan los requisitos como base para crear casos de prueba que utilizan para ejecutar el software bajo condiciones específicas y conocidas para detectar defectos y proporcionar confianza de que el software funciona según lo especificado.

-Los redactores de la documentación son responsables de utilizar los requisitos como entradas en la creación de la documentación del usuario, incluidos los manuales de usuario / operaciones, los archivos de ayuda, las instrucciones de instalación y los materiales de capacitación, según sea necesario.

-Los gerentes de proyecto son responsables de la planificación, seguimiento y control del proyecto y guían al equipo de desarrollo de software a la entrega exitosa del software.

-El soporte técnico, también llamado operaciones o la mesa de ayuda, es responsable de interactuar con la comunidad de usuarios para respaldar el software una vez que se ha implementado en el campo.

-La gestión de cambios del producto, que puede tomar la forma de una junta de control de cambios (CCB), es responsable de revisar los cambios propuestos a los requisitos, analizar sus impactos, aprobar / desaprobar los cambios y garantizar que los cambios aprobados son implementados y validados.

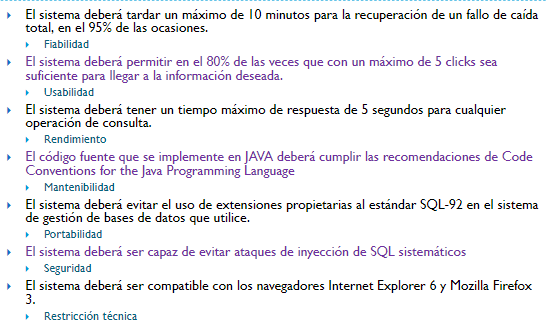
***Objetivos: ¿Qué quieres obtener?***

1. Necesito conocer tus requisitos antes de comenzar a diseñar el software
2. Lo primero ¿que quieres conseguir?
3. Estoy intentando que diseñes mi software
4. Quiero decir ¿qué quieres conseguir con tu software?
5. No se lo que puedo conseguir con mi software mientras no me digas que puede hacer el software
6. Trata de entender esta idea, el software puede hacer cualquier cosa para la que sea diseñado
7. ¿Puedes diseñarlo para que te diga mis requisitos?

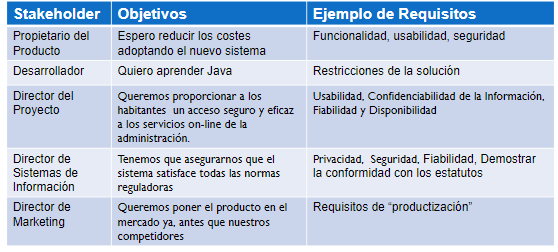
***Tipos de Requisitos***

-Funcionales: Indican lo que los usuarios necesitan que haga el sistema, funciones que el sistema proporciona (Un socio del club deportivo podrá reservar un máximo de 5 pistas a la semana y una única pista el mismo día a la misma hora).

-No funcionales: Características como, usabilidad, rendimiento… Especificaciones suplementarias y Vitales requisitos globales



***Requisitos y objetivos de los Stakeholder***

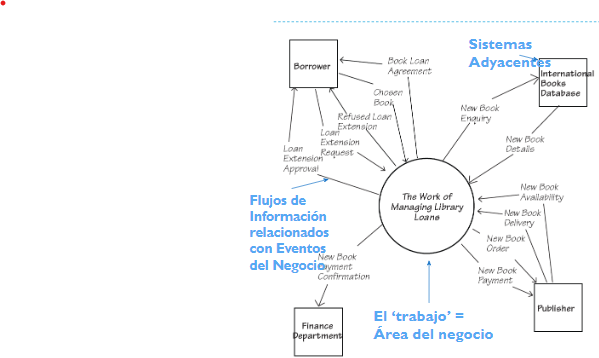


***¿Cuándo y cómo empezar? Reunión Inicial***

-Propósito del Proyecto y los acuerdos de la reunión inicial

-”El Trabajo”, área que se debe comprender ya que es afectada por la instalación del producto y se debe alcanzar despues del producto futuro

-Determinar el alcance, que se estudiara y que no. Serán necesarias las áreas de interés y los sistemas externos o adyacentes

***Diagrama de Contexto - Biblioteca*** 

-Muestra el “El Trabajo”

-Muestra los sistemas adyacentes y el flujo de información los sistemas

y “El Trabajo”

-El software se convierte con el tiempo en el “El Trabajo”

***Eventos del Negocio***

-Un Evento del Negocio tiene lugar fuera del alcance del trabajo, se da cuenta de que ocurre porque llega en el flujo de información

-La responsabilidad de lanzar el evento del negocio está fuera del control del trabajo. Tienen fecha de llegada o una hora. Se necesitan conocimientos. Los stakeholders lo conocen

***Eventos del Negocio y Casos de Uso del negocio***

-Para cada evento hay una BUC, casos de uso del negocio.

-BUC, es una unidad de funcionamiento cuyos pasos han sido obtenidos de las ayudas de los stakeholders

-El producto a desarrollar contribuirá al trabajo que realiza el BUC

-Organización, individuo, sistema informático o elemento tecnológico, o combinación de estos

-Tipos de sistemas adyacentes:

Activos: lo inician seres humanos con un objetivo en mente

Autónomos: comunicación por medio de flujos de una dirección

Cooperativos: comportamiento esperado, comunicación pregunta-respuesta

***Claves en la Ingeniería de Requisitos***

-**Descripción** del **problema** frente a la descripción de la **solución**: Esta distinción es **útil** para **comunicarse con los clientes y para evaluar diferentes soluciones de diseño**. Sin embargo, es **imposible** hacer esta distinción de **forma** totalmente **limpia porque** el problema y la solución **interactúan**.

-**Qué contra Cómo**: **"qué" debe hacer un sistema, sin decir "cómo" debe hacerlo**. El motivo es **evitar** el **"sesgo de solución" en la declaración del problema**, **no** debemos **escribir** **una declaración del problema** de manera tal que **sugiera ciertas soluciones o excluya otras**. Se puede **preferir algunas soluciones sobre otras, "lo que" versus "cómo"** es demasiado **simplista** para la mayoría de los propósitos.

-**Dominio de aplicación frente a dominio de máquina**: Estos mundos **se superponen de forma limitada**, y es esta superposición la que **nos permite tomar los requisitos de dominio de la aplicación que nos interesan** y **traducirlos** en relaciones **entre entradas y salidas** que el software puede controlar.

-**Requisitos funcionales y no funcionales**: Los requisitos **funcionales capturan las funciones que debe realizar un sistema**, mientras que los requisitos **no funcionales** capturan **propiedades generales** sobre el sistema, a menudo también se denominan "cualidades del sistema" o "idades".

-**Ingeniería de sistemas vs. Ingeniería de software**: La ingeniería de requisitos se aplica tanto a la ingeniería de sistemas como a la de software. Para la **ingeniería de software**, normalmente se supone que la **forma** en que el software **interactúa con el mundo es fija**, **utilizando tipos estandarizados de dispositivos de entrada y salida**. Para la **ingeniería de sistemas**, no se hacen tales suposiciones: la tarea es **diseñar un sistema** completo, del cual el **software es solo un componente**.

-**Clientes frente a usuarios**: el término **"partes interesadas "** para **indicar** en términos más amplios **a todos los diferentes grupos de personas** que pueden verse **afectadas** por un nuevo sistema y, por lo tanto, que **pueden tener requisitos que deben considerarse**. Dos subgrupos importantes son clientes y usuarios. Los **clientes** son los responsables de **encargar** **un nuevo sistema**, mientras que los **usuarios** son las personas que **interactuarán con el sistema** una vez que se instale. Sólo en **casos especiales** son la **misma persona o personas**.

-**Descripciones indicativas y optativas**: Al **describir un problema**, a menudo es necesario **hablar tanto de la situación actual como de la futura situación** prevista una vez que hayamos **diseñado una solución**. Una **declaración indicativa describe el mundo como está ahora**, mientras que una **declaración optativa** describe un estado de **cosas que nos gustaría lograr**. La distinción es importante porque tenemos que entender qué cosas se pueden asumir sobre el mundo y qué cosas se requieren para que el software produzca. **Las propiedades del dominio de aplicación son indicativas, mientras que los requisitos son optativos.**

-**Verificación frente a validación**: La **verificación es** el proceso de determinar que un **programa cumple con su especificación**, mientras que la **validación** es la tarea de **asegurarse** de que el sistema **abordará el problema correcto** en el mundo real. Esta cita captura la esencia de la distinción, pero recuerde que **la distinción sólo tiene sentido cuando se considera el papel de una especificación.**

-**Captura frente a sintetizar los requisitos**: **Las partes interesadas a menudo no saben lo que quieren o lo que es posible**. Es posible que **no sepamos exactamente quiénes serán los clientes y los usuarios** para algunos sistemas. En estas circunstancias, es **erróneo pensar que los requisitos están "ahí fuera" listos para ser capturados**. Más bien, **necesitan ser negociados o incluso inventados**. Pero esto no significa que simplemente los inventemos. En su lugar, **sintetizamos los requisitos en función de nuestra mejor comprensión del problema** que estamos tratando de resolver, junto con estimaciones razonables para las incógnitas.