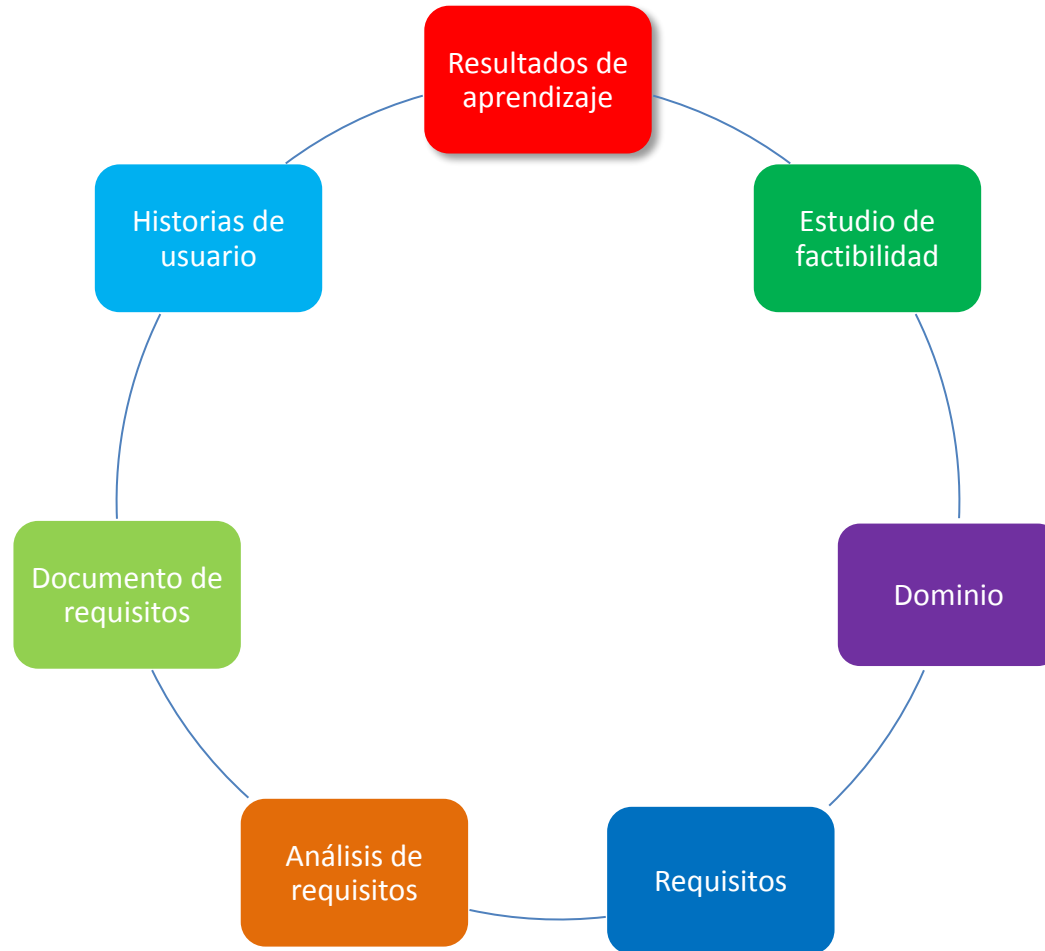


Análisis de Sistemas

CIF 5555

2022-1



ANÁLISIS DE REQUISITOS

Resultados de aprendizaje

- El alumno debería:
 - Aplicar el proceso de Ingeniería de Requisitos para la elicitación de las necesidades de los clientes y usuarios

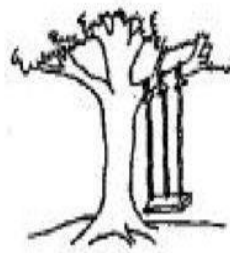
Análisis de Requisitos

Unidad 2. Ingeniería de Requisitos

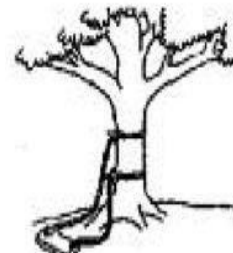
El vaivén de los requisitos



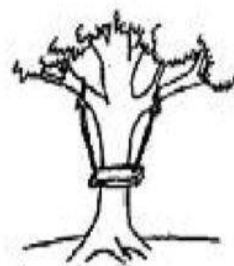
**As proposed
by the project
sponsor.**



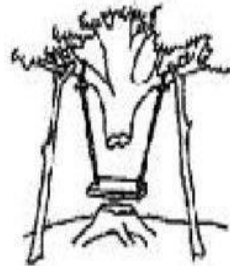
**As specified
in the project
request.**



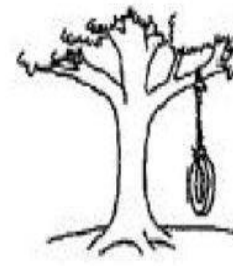
**As designed
by the senior
architect.**



**As produced
by the
engineers.**



**As installed at
the user's
site.**



**What the
customer
really wanted.**

En el quehacer cotidiano, esto se hace a menudo

Estudio de factibilidad

- Etapa preliminar para determinar si se tendrá éxito o no al construir el software
- Se basa en
 - Una breve descripción del sistema de software y las necesidades del usuario.
 - Un análisis para entender si tiene sentido hacer el producto:
 - Lo que ya existe
 - ¿Se venderá el producto?
 - Un análisis técnico para entender si es factible

Estudio de factibilidad

- Se basa en la evaluación de los costos y beneficios de una posible actividad productiva.
- Viabilidad tecnológica
 - Herramientas de implementación (software, bibliotecas, ...)
 - Soluciones algorítmicas y arquitectónicas
 - Hardware
 - Proceso (prototipos, proyectos de exploración, investigación ..)
- Aspectos económicos y de mercado
 - Comparación entre el mercado actual y futuro
 - Costo de producción, retorno de la inversión

Actividades de análisis de requisitos

- Estudiar y definir el problema a resolver
 - Para entender lo que hay que hacer
 - Para documentar lo que se necesita hacer
 - Para negociaciones cliente / proveedor

qué
no cómo

Implicaciones económicas y de calidad

- Las causas del abandono según el Grupo Standish
 1. Requisitos incompletos
 2. Baja participación de los usuarios
 3. Falta de recursos
 4. Expectativas no realistas
 6. Cambios en las especificaciones y requisitos
 - [.....]
 10. Ignorancia tecnológica

Cuánto “cuesta” corregir un error en los requisitos [Bohem]

FASE	COSTO
Análisis de requerimientos	1
Diseño	5
Codificación	10
Prueba de unidad	20
Examen de ingreso	50
Operación	100

Producto de las actividades de análisis

- Documento y / o modelo analítico de
 - Descripción de dominio
 - Descripción de los requisitos

- Opcionalmente también
 - Manual de usuario
 - Casos de prueba

- (a menudo en paralelo)

Dominio

- Dominio
 - El campo de aplicación del producto
- Antes incluso de conocer a los clientes, el equipo de analistas debe adquirir conocimientos sobre el dominio de la aplicación.
 - Para hacer las preguntas correctas: piense en el software de transacciones financieras
 - Para comprender la sutil diferencia que puede existir entre términos que sólo aparentemente son sinónimos: piense en soporte, poste, soporte, viga en ingeniería civil.

Dominio

- Para adquirir conocimientos y definir el dominio:
 - Se construye un **glosario**, el equipo de analistas lo construye a medida que estudian el dominio y luego se enriquece a medida que se encuentran nuevos términos
 - Se definen un **modelo estático** y un **modelo dinámico**
- El conocimiento del dominio evoluciona después de reunirse con los clientes, usuarios, etc.
- Proceso iterativo

Requisito

■ Requisito

- Una condición o capacidad necesaria para que un usuario resuelva un problema [..]
- Una condición (capacidad) que debe ser satisfecha (poseída) [..] por un sistema [..] para cumplir con un contrato [..]

[Glosario de IEEE]

■ En otras palabras:

- Una propiedad que debe ser garantizada por el sistema para satisfacer la necesidad de un usuario.
 - Funcionalidad, calidad, ...

Adquirir conocimiento sobre dominio y requisitos: Entrevistas

- Los miembros del equipo de analistas se encuentran con los miembros de la organización del cliente.
- Se procede con entrevistas que pueden ser
 - estructuradas
 - no estructuradas
- Ambas tienen ventajas y riesgos
- Difícil de realizar una buena entrevista

Adquirir conocimiento sobre dominio y requisitos: Otras técnicas

- Cuestionarios de opción múltiple para todos los miembros relevantes de la organización del cliente
- Construcción de prototipos
- Observación de futuros usuarios en el trabajo
- Estudio de documentos

Adquirir conocimiento sobre dominio y requisitos (funcionales) : Casos de uso

- Los casos de uso son otra forma de capturar los requisitos
- El caso de uso es una forma en que un usuario puede usar el producto.
- Se presenta al cliente junto con el resultado esperado y se esperan comentarios.
- Los casos de uso deben incluir no solo la secuencia correcta de eventos, sino también el comportamiento inesperado: **las excepciones.**

Categorías de requisitos : Requisitos funcionales

- Describen las funcionalidades que debe realizar el sistema
 - En términos de:
 - acciones que debe realizar el sistema
 - cómo reacciona el sistema de software a tipos específicos de entradas
 - cómo se comporta en situaciones particulares.
- Tradicionalmente los requisitos a los que se les da más valor
- Ejemplos
 1. El sistema de software debe proporcionar un visor adecuado para los documentos almacenados.
 2. El usuario debe poder buscar sea en todo el conjunto de bases.de datos que en un subconjunto de ellos
 3. Cada nuevo pedido debe estar asociado con un identificador único (Order_ID)

Categorías de requisitos : Requisitos no funcionales

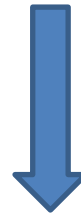
- también conocido como requisitos de calidad
- Describen las propiedades del sistema de software en relación con ciertos servicios o funciones y también pueden estar relacionados al proceso:
 - características de calidad como:
 - eficiencia, confiabilidad, seguridad, usabilidad, interfaz, seguridad, robustez, etc.
 - características del proceso de desarrollo:
 - estándares de proceso, uso de entornos CASE, lenguajes de programación, métodos de desarrollo, etc.
 - características externas:
 - interoperabilidad con sistemas de otras organizaciones, limitaciones legislativas, etc.
 - requisitos físicos (hw)

Ejemplos de requisitos no funcionales

1. El tiempo de respuesta del sistema al ingresar la contraseña de usuario debe ser menor a 10 segundos
2. Los documentos del proyecto (entregables) deben cumplir con el estándar XYZ-ABC-12345
3. El sistema de software no debe revelar a sus operadores ninguna información personal relacionada con los clientes, excepto nombre e identificador

Requisitos funcionales vs requisitos no funcionales

- Es bueno mantenerlos separados
- Ejemplo:
 - El sistema debe validar el pin ingresado por el cliente en 3 segundos



- El sistema debe validar el pin ingresado por el cliente
- La validación debe completarse en 3 segundos

Ejemplos de análisis de problemas

■ Monitoreo de tráfico:

- **Problema:** El usuario se retrasa para ir a trabajar y necesita ayuda
- Causa A: El tráfico es impredecible (¿predeciblemente alto si hay retrasos repetidos?)
- Causa B: el usuario no está familiarizado con la ruta
- Causa C: el usuario tiene la costumbre de empezar tarde

■ Automatización de restaurantes:

- **Problema:** El restaurante funciona con pérdidas y los clientes no están contentos
- Causa A: El personal tiene problemas de comunicación y deberes administrativos, lo que genera ineficiencias.
- Causa B: el menú no coincide con el gusto del cliente más probable
- Causa C: el personal se está portando mal con los clientes o administra mal los suministros de alimentos

■ Vigilancia de la salud:

- **Problema:** El usuario lleva un estilo de vida poco saludable y tiene problemas de salud.
- Causa A: el usuario está intentando estar activo pero no está seguro del nivel suficiente de actividad
- Causa B: el usuario intenta estar activo pero está demasiado ocupado con otras tareas
- Causa C: el usuario no puede encontrar soluciones asequibles para un estilo de vida activo
- Causa D: el usuario no duerme bien
- Causa E: el usuario es consciente de los problemas pero no está motivado para actuar

Ejemplo de problema: acceso a un hogar seguro

■ Problema detectado:

- llaves físicas inconvenientes o intrusión no deseada (más: operar dispositivos domésticos y minimizar los gastos de manutención)

■ Análisis de las causas:

- El usuario se olvida de cerrar la puerta o apagar los dispositivos
- El usuario pierde la llave física
- Incapacidad para rastrear el historial de accesos
- Incapacidad para operar de forma remota la cerradura y los dispositivos
- El intruso obtiene acceso

■ Requisitos del sistema: según las causas seleccionadas

Heurísticas para encontrar escenarios

- Hacerse a sí mismo o al cliente las preguntas siguientes
 - Cuáles son las tareas principales que el sistema necesita realizar
 - Qué datos en el sistema serán creados, almacenados, cambiados, eliminados o agregados por el actor
 - Qué cambios externos necesita el sistema conocer
 - Qué cambios o eventos necesitará que el actor o el sistema esté informado

Heurísticas para encontrar escenarios (cont.)

- No confiarse sólo en los cuestionarios...
- Insistir en la observación directa de las tareas si el sistema ya existe (ingeniería de interfaces o re-ingeniería)
 - Consultar para hablar con el usuario final, no solamente con el contratista del software
 - Esperar por resistencias y tratar de superarlas
- Prototipo temprano para levantar escenarios más complejos

Ejercicio: ATM

Considere un sistema para retiro de efectivo automatizado (cajeros automáticos). Para poder utilizar el sistema el usuario debe tener una tarjeta magnética. El sistema debe proporcionar las funciones de giro, saldo, estado de cuenta. El sistema debe estar disponible para personas con discapacidad, debe garantizar un tiempo de respuesta inferior por minuto, y debe desarrollarse en la arquitectura X86. Las operaciones de giro deben requerir autenticación a través de un código secreto almacenado en la tarjeta. El sistema bancario debe ser fácilmente ampliable y adaptable a las necesidades futuras.

Enfoque que se verá en el curso

- Basado en documentos en lenguaje natural
 - Glosario
 - Especificación de requisitos
 - funcional
 - calidad (no funcional)

- Basado en lenguajes formales (modelos / gráficos)
 - UML
 - Modelo de dominio
 - Casos de uso
 - Requisitos funcionales

Documento de requisitos

- El documento de requisitos es un contrato entre el desarrollador y el cliente
- Especificar qué debe hacer el producto y qué restricciones debe cumplir
- También suele especificar una fecha límite para la entrega del producto

Documento de requisitos en 5 pasos

1. Elicitación
2. Elaboración
3. Negociación
4. Validación
5. Gestión

Elicitación de requisitos (otra presentación)

- Entrevistas
 - Estructurado o no estructurado
- Cuestionarios escritos
 - Múltiples opciones,....
- Observación de futuros usuarios en el trabajo
- Estudio de documentos
- Producción de prototipos
- Casos de uso
- Con clientes, usuarios, directores de marketing

Elaboración

- Requisitos ampliados y refinados
- Definición del documento de requisitos
 - Basado en el uso del lenguaje natural
 - Descripción de dominio

Estructura del documento de requisitos

Propuesto por Sommerville e inspirado en el estándar IEEE / ANSI 830-1993 sobre documentos de requisitos.

■ Introducción

- Por qué el sistema es deseable y cómo encaja en los objetivos más generales del cliente

■ Glosario

- Los términos y conceptos técnicos utilizados

■ *Definición* de los requisitos funcionales

- Los servicios solicitados

■ *Definición* de los requisitos no funcionales

- Las limitaciones operativas del sistema y las del proceso de desarrollo.

■ Arquitectura

- Estructuración en subsistemas (a los que remitir los requisitos)

Estructura del documento de requisitos

- Especificación de requisitos de sistema y software
 - Especificación detallada de requisitos funcionales
- Modelos abstractos del sistema
 - Modelos formales o semiformales: cada uno ilustra solo un punto de vista: control, datos, funciones
- Evolución del sistema
 - Previsión de cambios posteriores (por ejemplo, de HW o requisitos)
- Apéndices
 - Identificación y posible descripción de la plataforma de hardware
 - Requisitos de la base de datos
 - Manual de usuario, planes de prueba
- Índices
 - lista de términos, con punteros a los requisitos que los utilizan

Glosario

- Definición de los términos clave del dominio
- Herramienta para comprender y documentar el dominio
- Pero no solo:
 - Validación de requisitos

Validación de requisitos

Validación de un documento ya estructurado

- Deskcheck
 - Walkthrough
 - lectura secuencial de documentos
 - Inspección
 - Lectura "estructurada" de documentos
 - Ejemplos:
 - Búsqueda de eliminaciones, distorsiones, generalizaciones.
(Noam Chomsky 1975)
- Uso de herramientas de análisis del lenguaje natural
- Prototipos

Propiedades del documento de requisitos

- Defectos a evitar
 - Omisiones
 - Incumplimiento de un requisito (falta de cumplimiento)
 - Inconsistencias
 - Contradicción entre varios requisitos o requisitos con respecto al entorno operativo
 - Ambigüedad
 - Requisitos con múltiples significados
 - Sinónimos y homónimos
 - Diferentes términos con el mismo significado y los mismos términos con diferentes significados
 - Presencia de detalles técnicos
 - Redundancia
 - Puede haber redundancia, pero solo entre diferentes secciones.

Completitud, Consistencia, Claridad, y Correctitud

- La validación de requisitos es un paso crítico en el proceso de desarrollo, normalmente después de la ingeniería de requisitos o el análisis de requisitos. Incluso en la entrega (nivel de aceptación del cliente)
- **Correctitud:** Los requisitos deben representar la visión del cliente
- **Completitud:** Todos los posibles escenarios, en los que el sistema pueda ser usado, son descritos, incluyendo comportamiento excepcional por el usuario o el sistema
- **Consistencia:** No hay dos requisitos que se contradicen el uno al otro
- **Inequívoco:** Un requisito no puede ser interpretado en dos maneras mutuamente exclusivas

Realismo, Verificabilidad, y Trazabilidad

- **Realismo:** Los requisitos pueden ser realizados y entregados
- **Trazabilidad:** Cada función de sistema puede ser localizada en un conjunto equivalente de requisitos funcionales
- **Verificabilidad:** Cada requisito del sistema puede ser probado usando algunas pruebas repetibles
 - *La aplicación tendrá una buena interfaz de usuario* – No se define qué es bueno...

Incompletitud

- *El sistema debe permitir a los usuarios buscar por título, autor o ISBN.*
- *¿Qué significa para los CD-ROM?*
 - Puede que no tengan un ISBN
 - Solo se aplica a libros
 - Imagínese si construye el sistema con este requisito sin tener en cuenta los CD-ROM.
- Por supuesto, no podemos escribir requisitos universales, pero podemos intentar ser lo más completos posible.

Inequívoco

- Ejemplo: "Todos los archivos están controlados por un bloque de control"
- Interpretaciones:
 - Un solo bloque de control controla todos los archivos
 - Cada archivo tiene su propio bloque de control específico
 - Cada archivo está controlado por un bloque de control específico; algunos bloques de control pueden controlar más de un archivo
- El lenguaje natural es intrínsecamente ambiguo

Técnica del lemario

- Términos del glosario, con punteros a los requisitos que los nombran

- Facilita la búsqueda de
 - Inconsistencias
 - Sinónimos
 - Homónimos
 - Redundancias

Ejemplos de requisitos bien dispuestos

El <sistema> debe <funcionalidad> / <propiedad>

- El sistema debe gestionar todas las cajas registradoras de la tienda (no más de 20)
- El sistema debe imprimir el resumen de los recibos del día
- Al final del día, el sistema debe enumerar qué artículos se reintegrarán en función de las ventas
- El tiempo de respuesta a cualquier pregunta debe ser menos de 3 segundos
- ¿¿¿Margen de mejora ???

Negociación

- Cambios en la lista de requisitos
- División en clases (por ejemplo, MoSCoW)
 - Debe tener (Requisitos obligatorios)
 - Esencial para el cliente
 - Debería tener (requisitos deseables)
 - No es necesario, pero es útil
 - Podría tener (Requisitos opcionales)
 - Relativamente útil, se hará si hay tiempo.
 - Quisiera tener
 - Negociable para versiones posteriores

Gestión de los requisitos:

1. identificación

- Identificador único
 - clave en una base de datos
- Número secuencial (1,2,3,4.)
- Número secuencial basado en la estructura del documento (2.4.7)
- Pares <CATEGORÍA, NÚMERO>

Gestión de los requisitos:

2. atributos (ejemplo tomado de RUP)

- Estado
 - Propuesta, aprobada, rechazada, incorporado
- Beneficios
 - Importancia, tipo MoSCoW
- Esfuerzo en hombre / días
- Riesgo
 - Evaluación de viabilidad técnica
- Estabilidad
- Versión de destino
 - Para un desarrollo incremental

Gestión de los requisitos:

2. trazabilidad

- Requisitos vs
 - Componentes del sistema
 - módulos
 - prueba

- Herramientas CASE

Requisitos y aspectos contractuales

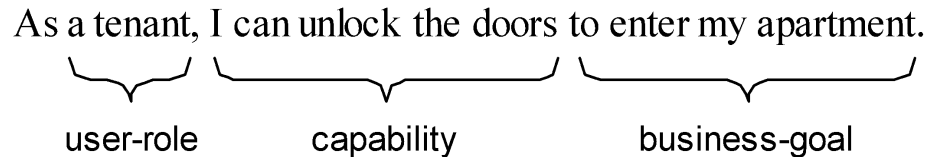
- El documento de requisitos normalmente precede a la firma del contrato y es parte integral del mismo
- Si al momento de la firma del contrato no es posible tener un documento definitivo, es recomendable planificar la renegociación del contrato
- En caso de licitación, el cliente puede presentarlo como parte de las especificaciones técnicas

SRS

- **29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering**
- Este documento contiene disposiciones para los procesos y productos relacionados con la ingeniería de requisitos para sistemas y productos y servicios de software a lo largo del ciclo de vida. Define la construcción de un buen requisito, proporciona atributos y características de los requisitos, y discute la aplicación iterativa y recursiva de los procesos de requisitos a lo largo del ciclo de vida.
- Este estándar reemplaza ISO/IEC/IEEE 29148:2011(E) (que reemplazó IEEE 830-1998, IEEE 1233-1998, IEEE 1362-1998)

Requisitos como historias de usuario

As a tenant, I can unlock the doors to enter my apartment.



user-role capability business-goal

- Expresado en términos de objetivos y capacidades del usuario en lugar de características del sistema
 - Escrito por el cliente o usuario, no por el desarrollador
 - El esfuerzo de desarrollo para implementar una historia se estima de inmediato
 - Las pruebas de aceptación se escriben cuando se identifica la historia
 - Requisitos identificados solo para la siguiente iteración, no para todo el proyecto
-
- https://es.wikipedia.org/wiki/Desarrollo_guiado_por_comportamiento

Requisitos de ejemplo de historia de usuario

Automatización de restaurantes

Identificador	Historia de usuario	Tamaño
REQ-1	Como un anfitrión, puedo tomar una solicitud de asiento que incluya información de la fiesta del cliente, colocarla en la fila de asientos y tener una mesa asignada o un tiempo de espera estimado.	10 puntos
REQ-2	Como camarero, puedo ingresar el pedido del cliente.	5 puntos
REQ-3	Como camarero, puedo agregar instrucciones especiales a un pedido a pedido del cliente.	2 puntos
REQ-4	Como camarero, puedo notificar al chef del pedido sin tener que caminar hasta la cocina.	2 pts
REQ-5	Como mesero, puedo ver la factura del cliente e ingresar su información de pago.	7 pts
REQ-6	Como mesero, se me notificará cuando se haya completado un pedido.	2 pts
REQ-7	Como chef, puedo ver la cola de pedidos esperando ser preparados.	3 pts
REQ-8	Como chef, puedo marcar pedidos como "En preparación" y "Completo".	2 pts
REQ-9	Como chef, puedo modificar el menú para que ciertos platos estén disponibles o no disponibles si los suministros son limitados.	6 pts
REQ-10	Como chef, puedo ajustar y actualizar el inventario de suministros para informar al gerente de cualquier ingrediente que falte.	7 pts

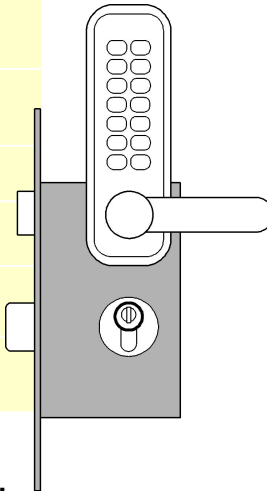
- Después del análisis de requisitos:

Identificador	Historia de usuario	Tamaño
REQ-2a	Como camarero, puedo ingresar pedidos en diferentes momentos para los clientes en la misma mesa.	7 puntos
REQ-2b	Como camarero, puedo ingresar diferentes platos en diferentes momentos para el mismo cliente.	5 pts
REQ-2c	Como camarero, puedo introducir platos de acompañamiento después de que se sirva el plato principal.	2 pts

Requisitos de ejemplo de historia de usuario

Acceso seguro a casa

Identificador	Historia del usuario	Tamaño
REQ-1	Como usuario, puedo estar seguro de que las puertas estarán cerradas por defecto.	4 puntos
REQ-2	Como usuario, podré desbloquear las puertas con una llave válida.	7 puntos
REQ-3	Un intruso no podrá abrir las puertas adivinando una llave válida; el sistema se bloqueará cuando detecte un "ataque de diccionario".	7 puntos
REQ-4	Como usuario, puedo estar seguro de que las puertas se bloquearán automáticamente en todo momento.	6 pts
REQ-5	El teclado de la puerta estará retroiluminado cuando esté oscuro para mayor visibilidad.	3 pts
REQ-6	Cualquiera podrá cerrar las puertas a pedido.	3 pts
REQ-7	Como usuario, podré administrar cuentas de usuario adicionales.	10 pts
REQ-8	Como usuario, podré ver el historial de accesos a mi casa.	6 pts
REQ-9	Como usuario, podré configurar las preferencias de cómo se activarán mis dispositivos domésticos a mi llegada.	6 pts



- No tenga en cuenta las prioridades para las historias de usuario
 - La prioridad de la historia viene dada por su orden de aparición en la lista de tareas pendientes.
- Los puntos de tamaño estimado (última columna) se describirán más adelante
- Compare con los requisitos de estilo IEEE-830
 - https://en.wikipedia.org/wiki/Software_requirements_specification

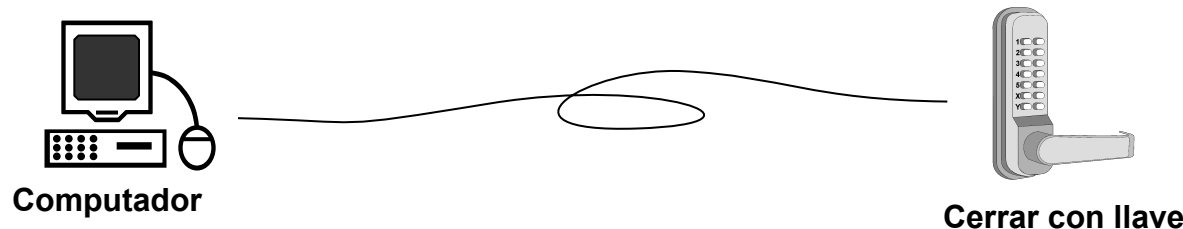
Análisis de requisitos

- El requisito REQ-3 establece que los intrusos no podrán tener éxito con un "ataque de diccionario", pero es necesario considerar muchos detalles y determinar muchos parámetros.
 - Qué distingue los errores del usuario de los "ataques de diccionario"
 - El número de intentos fallidos permitidos, en relación con un valor de umbral predefinido
 - El umbral será pequeño; por ejemplo, tres ☐ **Política del negocio**
 - ¿Cómo se relaciona el bloqueo mecánico con el estado "bloqueado"?
 - ¿Puede el usuario utilizar la llave mecánica cuando el sistema está "bloqueado"?
- El requisito REQ-5 establece que el teclado debe estar retroiluminado cuando esté oscuro
 - ¿Es rentable detectar la oscuridad en lugar de mantenerlo siempre encendido?
- ...

El análisis de requisitos no debe ser exhaustivo, pero ninguno debe evitarse.

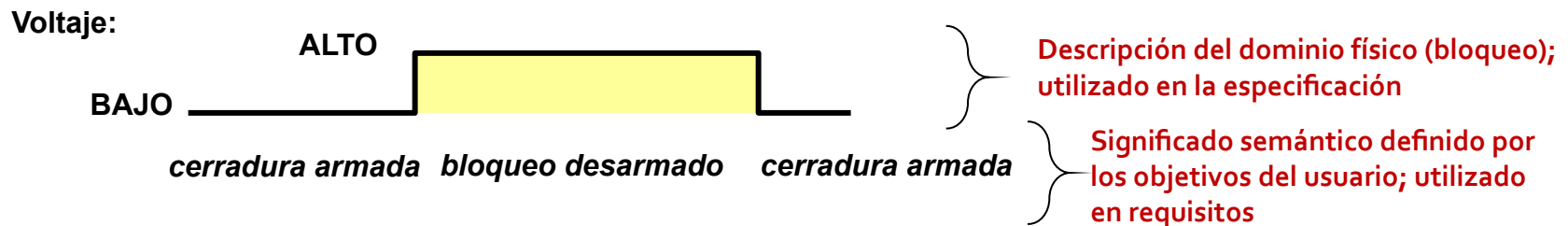
https://es.wikipedia.org/wiki/Ataque_de_diccionario

Dominio del problema: Cómo funciona la cerradura electrónica



Es posible que necesitemos descripciones / modelos separados de puerta frente a cerradura.

El estado de la puerta es lo que le importa al usuario; bloquear es una forma de lograrlo.



El comportamiento del sistema a ser determinado no solo por las acciones del usuario, pero también por el contexto ("situación").

Por ejemplo, ¿qué pasa en caso de corte de energía?

- Por defecto armado
- Por defecto desarmado (p. Ej., Salida de incendios)

Tarea del analista: Tres descripciones

El requisito

Qué quiere el usuario:

Cuando se ingresa una clave válida Y Desbloquear presionado, abrir la cerradura;

Bloquear automáticamente después de un período de tiempo.



El dominio del problema

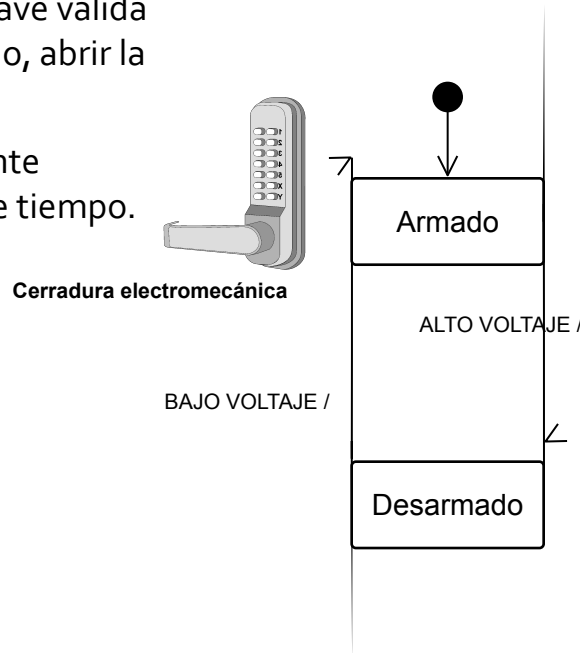
Cómo se comporta el dominio del problema



La especificación

Qué hará el software futuro (en la interfaz):

Si el número ingresado coincide con uno de los números almacenados Y Botón-1 presionado, poner ALTO voltaje en el Puerto-salida-1;
Iniciar una cuenta regresiva del temporizador;
Cuando el temporizador expira, poner BAJO voltaje en el Puerto-salida-1.



Inquietud:

No es obvio que esta sea la única o incluso la solución "correcta" al problema planteado por los requisitos.

