



# Ingeniería de Software

GCS - Requerimientos - Tablas de Decisión – DTE - Redes de Petri

# Contenidos

---

- »Gestión de la Configuración del Software (GCS)
- »Requerimientos
- »Ingeniería de Requerimientos
  - Estudio de Viabilidad
  - Obtención y análisis de requerimientos
  - Especificación de requerimientos
  - Validación de requerimientos
- »Gestión de Requerimientos
- »Técnicas de especificación
  - Estáticas
  - Dinámicas

2

# Gestión de la Configuración del Software (GCS)

---

## »¿Qué es Software?

Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación

»Gestión de Configuración es el proceso de identificar y definir los elementos en el sistema, controlando el cambio de estos elementos a lo largo de su ciclo de vida, registrando y reportando el estado de los elementos y las solicitudes de cambio, y verificando que los elementos estén completos y que sean los correctos.

»Es una actividad de autoprotección que se aplica durante el proceso del software.

3

# Gestión de la Configuración del Software (GCS)

»El resultado del proceso de Software se puede dividir en:

Programas (códigos y ejecutables)  
Documentos  
Datos

Elementos de la  
configuración (ECS)

Avanza el  
Proyecto

Control muy exhaustivo  
de esos cambios

ECS - Cambian  
constantemente

GCS

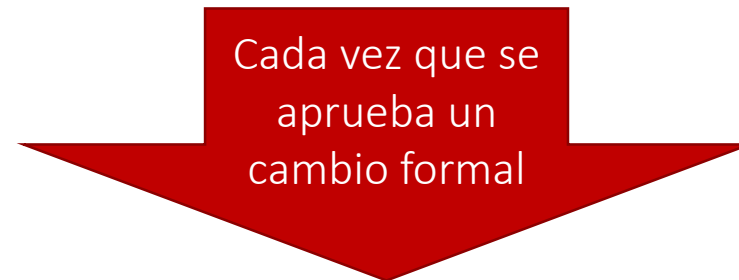
# Gestión de la Configuración del Software (GCS)

- »2) Plan del proyecto software
- »3)
  - a) Especificación de requerimientos del software
  - b) Prototipo ejecutable o en papel
- »4) Manual de usuario preliminar
- »5) Especificación de diseño:
  - a) Diseño preliminar
  - b) Diseño detallado
- »6) Listados del código fuente
- »7)
  - a) Planificación y procedimiento de prueba
  - b) Casos de prueba y resultados registrados
- »8) Manuales de operación y de instalación
- »9) Programas ejecutables
- »10) Manual de usuario
- »11) Documentos de mantenimiento
  - a) Informes de problemas del software
  - b) Peticiones de mantenimiento
  - c) Órdenes de cambios de ingeniería
- »12) Estándares y procedimientos de ingeniería del software

# Gestión de la Configuración del Software (GCS)

---

- »El cambio se puede producir en cualquier momento, las actividades de la GCS sirven para:
- Identificar el cambio
  - Controlar el cambio
  - Garantizar que el cambio se implemente adecuadamente
  - Informar del cambio a todos aquellos que puedan estar afectados



Línea Base

6

# Gestión de la Configuración del Software (GCS)

## »Línea Base

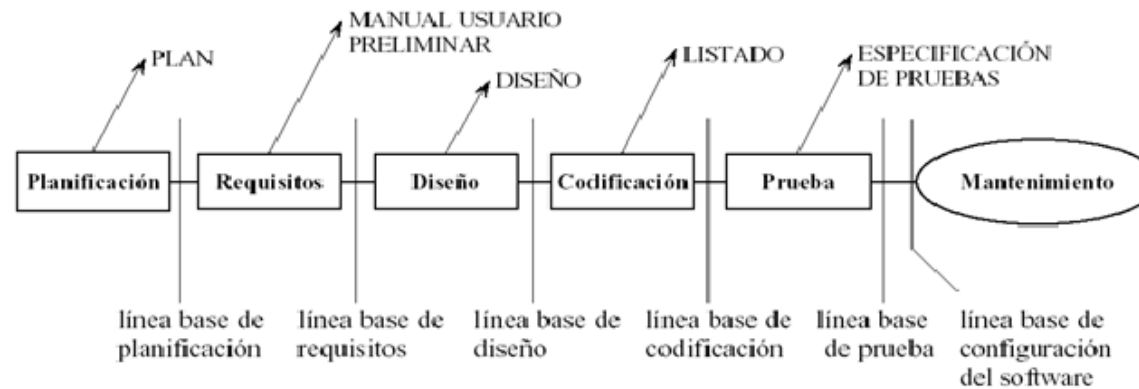
Una línea base es un concepto de GCS que nos ayuda a controlar los cambios

Definición de la IEEE

*Una especificación o producto que se ha revisado formalmente y sobre los que se ha llegado a un acuerdo, y que de ahí en adelante sirve como base para un desarrollo posterior y que puede cambiarse solamente a través de procedimientos formales de control de cambio*

En el contexto de la Ingeniería de Software:

*una línea base es un punto de referencia en el desarrollo del software que queda marcado por el envío de uno o mas ECS y su aprobación*



# Gestión de la Configuración del Software (GCS)

---

## »Importancia de la GCS

¿Cómo identifica y gestiona una organización las diferentes versiones existentes de un programa (y su documentación) de forma que se puedan introducir cambios eficientemente?

¿Cómo controla la organización los cambios antes y después de que el software sea distribuido al cliente?

¿Quién tiene la responsabilidad de aprobar y de asignar prioridades a los cambios?

¿Cómo podemos garantizar que los cambios se han llevado a cabo adecuadamente?

¿Qué mecanismo se usa para avisar a otros de los cambios realizados?

8



# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

1. *Identificación*
2. *Control de versiones*
3. *Control de cambios*
4. *Auditorías de la configuración*
5. *Generación de informes*

9

# Gestión de la Configuración del Software (GCS)

## »Proceso de la GCS

### 1- Identificación de los objetos en la GCS

*Nombre: cadena de caracteres sin ambigüedad*

*Descripción: lista de elementos de datos que identifican:*

Tipo de ECS (documento, código fuente, datos)

Identificador del proyecto

Información de la versión y/o cambio

ING - IC – CLASE N - 2016

Identificación  
Univoca

Año

Numero de clase

Ingeniería en  
Computación

Ingeniería de  
Software

10

# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 2 - Control de versiones

*Permite al usuario especificar configuraciones alternativas del sistema mediante la selección de versiones adecuadas (por ejemplo asociando atributos que la identifican)*

*Combinación de procedimientos y herramientas para gestionar las versiones de los ECS*

*Ejemplo de versiones*

Un programa puede contener los módulos 1-2-3-4-5

Una versión puede utilizar los módulos 1235

Otra versión puede utilizar los módulos 1245

Dos variantes de una misma versión

11

# Gestión de la Configuración del Software (GCS)

## »Proceso de la GCS

### 2 - Control de versiones

#### *Repositorio*

Se almacenan los archivos actualizados e históricos de cambio del proyecto.

#### *Versión*

Determina un conjunto de archivos

#### *Master*

Conjunto de archivos principales del proyecto

#### *Abrir rama – branch*

Bifurcación del máster para trabajar sobre dos ramas de forma independiente

#### *Desplegar – check-out*

Copia de trabajo local desde el repositorio.

#### *Publicar - Commit*

Una copia de los cambios hechos a una copia local es escrita o integrada sobre repositorio.

#### *Conflicto*

Problema entre las versiones de un mismo documento

#### *Cambio – diff*

Representa una modificación específica a un

#### *Integración – Merge*

Fusión entre dos ramas del proyecto

#### *Actualización – sync o update*

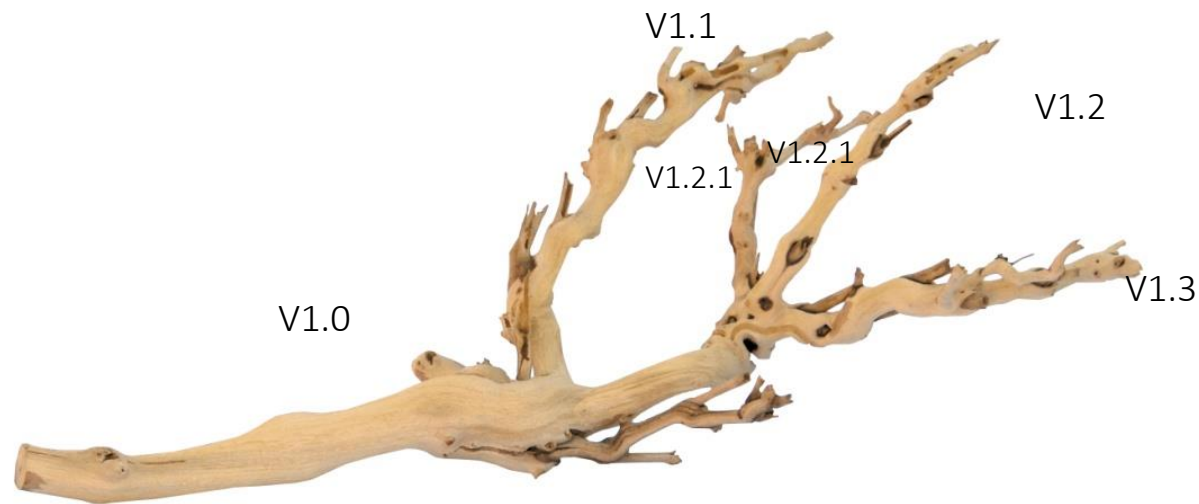
Integra los cambios que han sido hechos en el repositorio y las copias locales

12

# Gestión de la Configuración del Software (GCS)

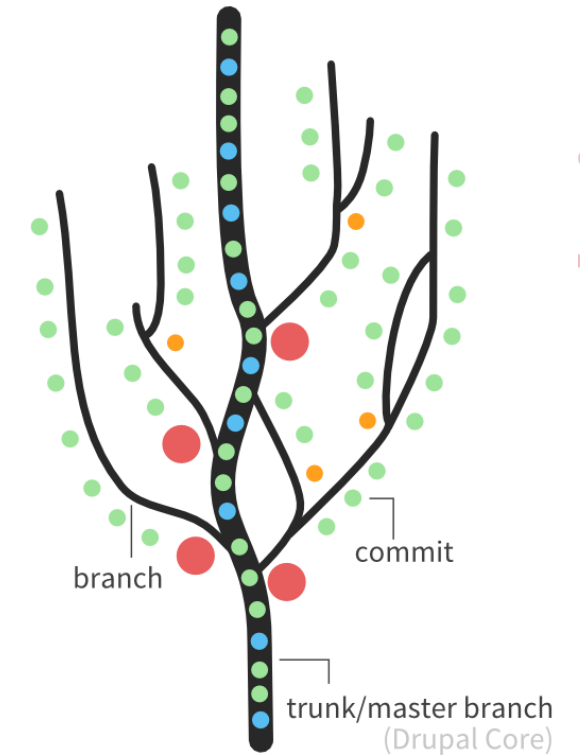
## »Proceso de la GCS

### 2 - Control de versiones



© Exo Terra - PT-3076

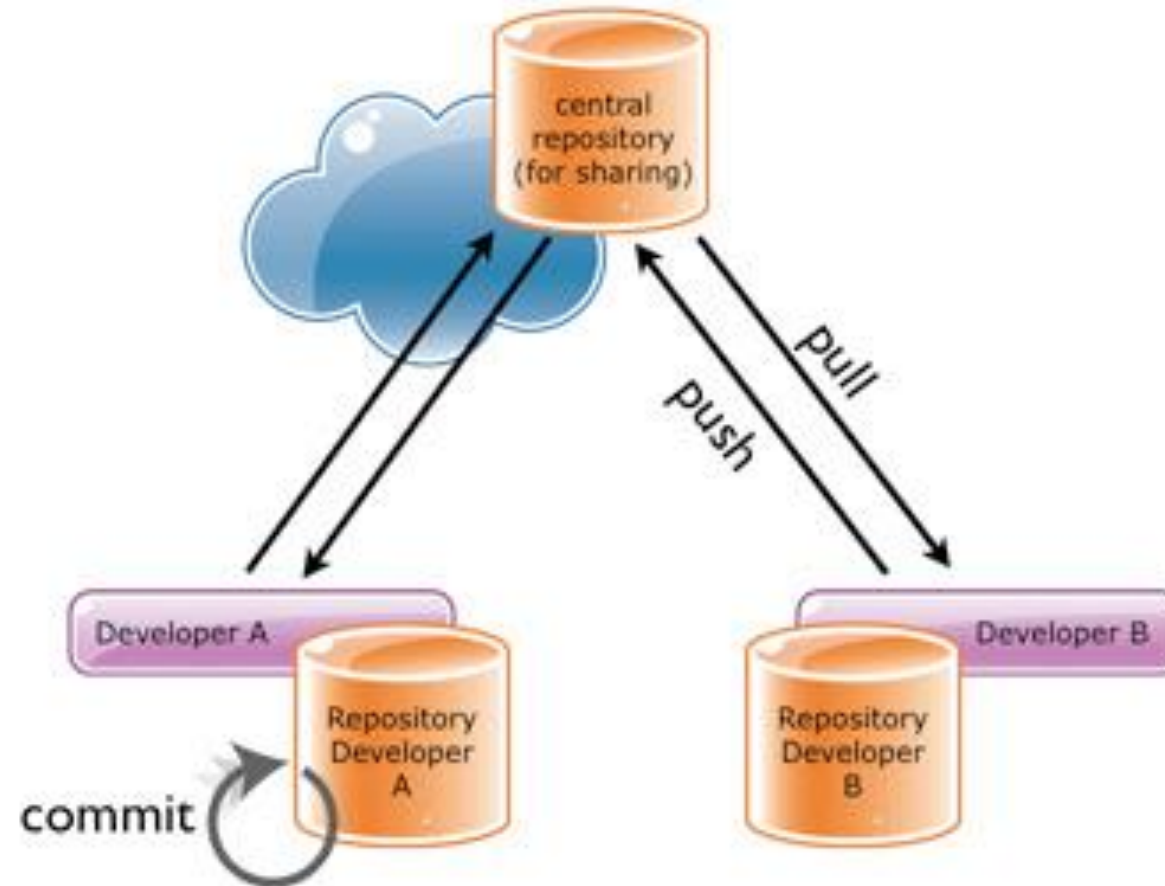
Fuente:



3

# Gestión de la Configuración del Software (GCS)

- »Proceso de la GCS
  - 2 - Control de versiones

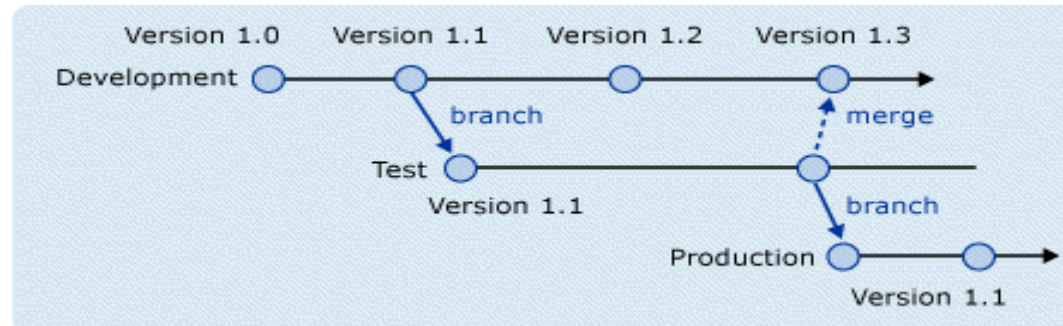
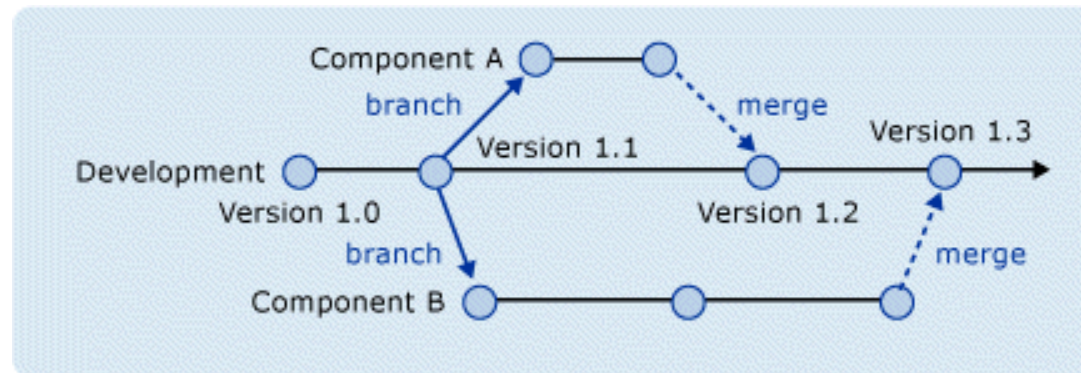


14

# Gestión de la Configuración del Software (GCS)

»Proceso de la GCS

2 - Control de versiones



15

# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 3 - Control de cambios

*A lo largo del proyecto los cambios son inevitables y el control es vital para el desarrollo del mismo*

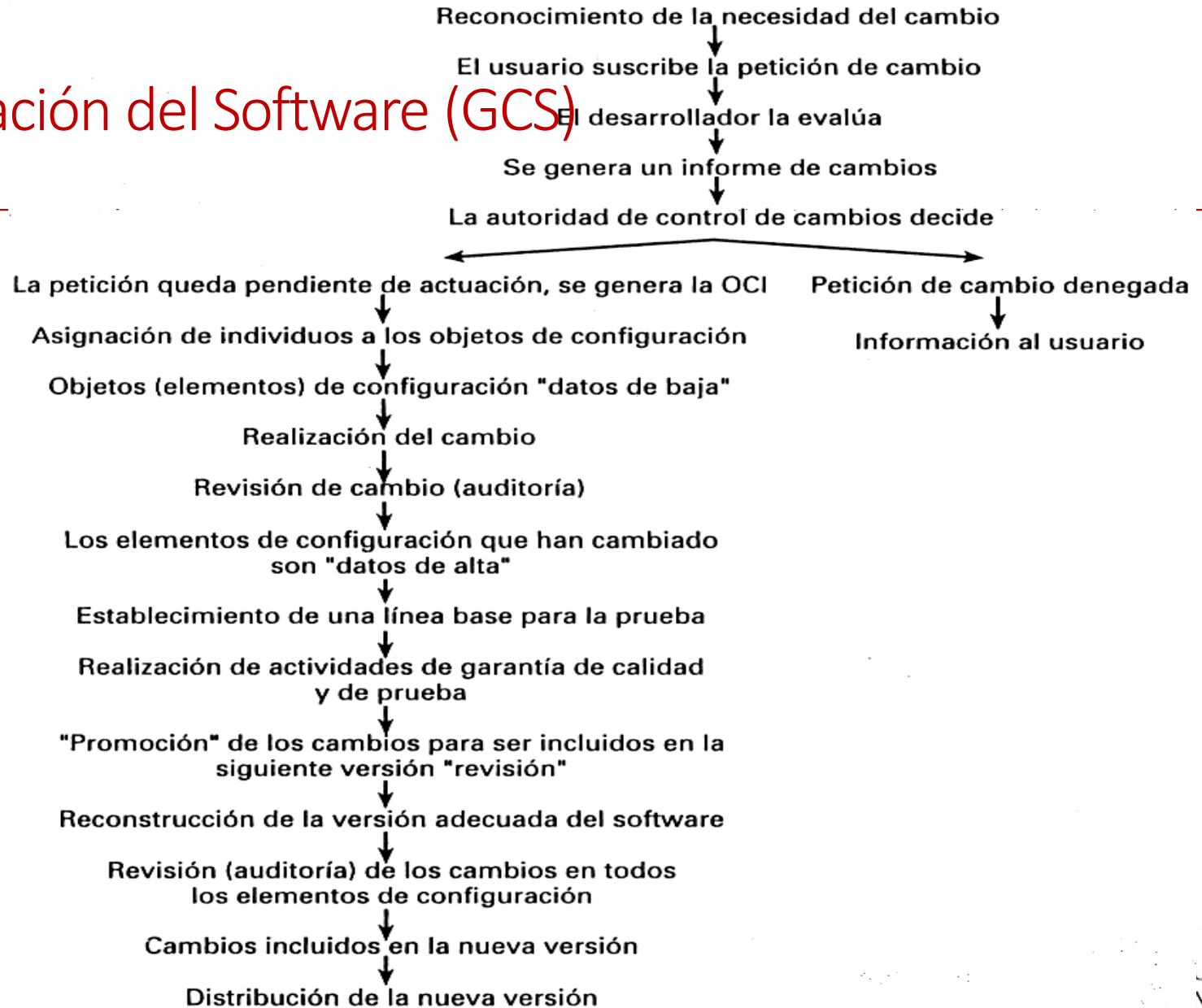
*Combina los procedimientos humanos y las herramientas adecuadas para proporcionar un mecanismo para el control del cambio*

16



# Gestión de la Configuración del Software (GCS)

- »Proceso de la GCS
- 3 -Control de cambios



17

# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 3 -Control de cambios

*La autoridad de control de cambios (ACC) evalúa:*

- ¿Cómo impactará el cambio en el hardware?
- ¿Cómo impactará el cambio en el rendimiento?
- ¿Cómo alterará el cambio la percepción del cliente sobre el producto?
- ¿Cómo afectará el cambio a la calidad y a la fiabilidad?

...

18

# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 4 - Auditoría de la configuración

*La identificación y el control de versiones y el control de cambio, ayudan al equipo de desarrollo de software a mantener un orden, pero sólo se garantiza hasta que se ha generado la orden de cambio.*

*Cómo aseguramos que el cambio se ha realizado correctamente*

Revisiones técnicas formales

Auditorías de configuración

19

# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 4 - Auditoría de la configuración responde:

*¿Se ha hecho el cambio especificado en la Orden de Cambio? ¿Se han incorporado modificaciones adicionales?*

*¿Se ha llevado a cabo una RTF para evaluar la corrección técnica?*

*¿Se han seguido adecuadamente los estándares de IS?*

*¿Se han reflejado los cambios en el ECS: fecha, autor, atributos?*

*¿Se han seguido procedimientos de GCS para señalar el cambio, registrarlo y divulgarlo?*

*¿Se han actualizado adecuadamente todos los ECS relacionados?*

20

# Gestión de la Configuración del Software (GCS)

---

## »Proceso de la GCS

### 5 - Generación de informes de estado de la configuración

*Responde*

¿Qué pasó?

¿Quién lo hizo?

¿Cuándo pasó?

¿Qué más se vio afectado?

21

*La generación de informes de estado de la configuración desempeña un papel vital en el éxito del proyecto*

# Requerimientos

22

# 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

»Definición IEEE-Std-610

1. Condición o capacidad que necesita el usuario para resolver un problema o alcanzar un objetivo
2. Condición o capacidad que debe satisfacer o poseer un sistema o una componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.
3. Representación documentada de una condición o capacidad como en 1 o 2.

23

# Requerimientos

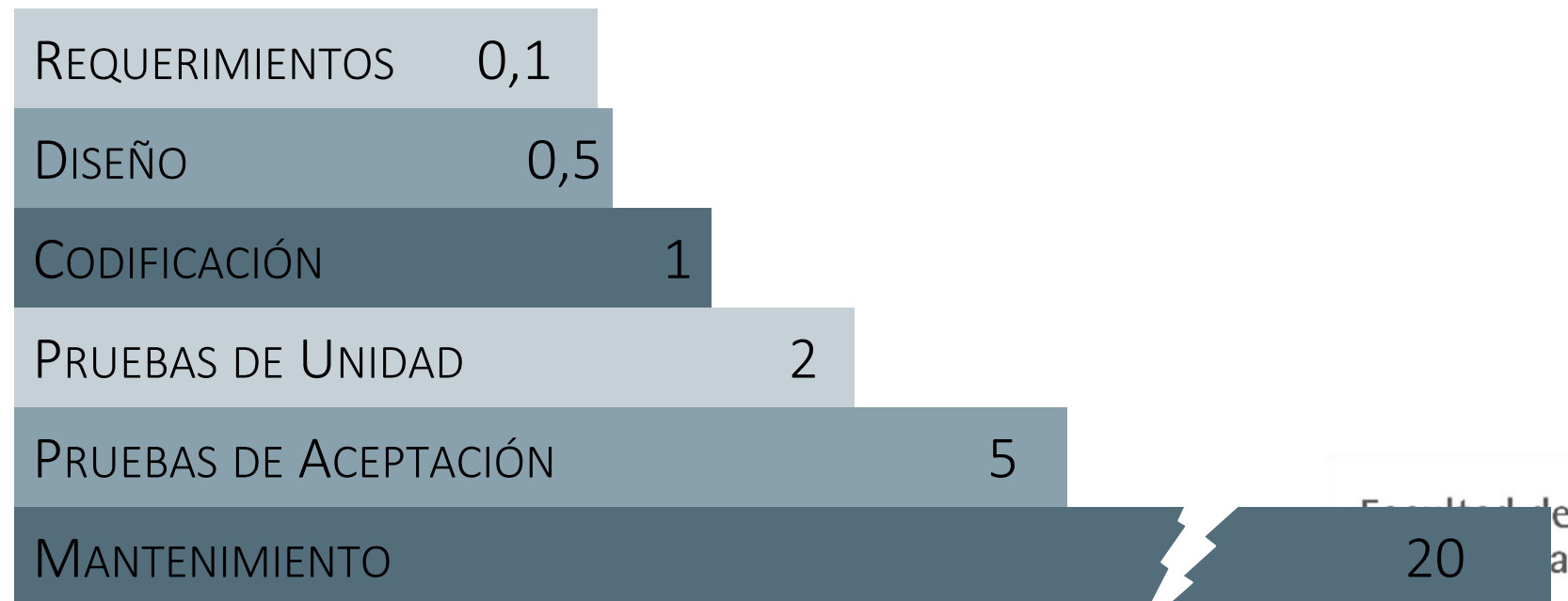
## » Impacto de los errores en la etapa de requerimientos

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

## » Solucionar el error de calculo en una formula compleja



24



# Ingeniería de Requerimientos

---

- »La ingeniería de requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en donde se describen las funciones que realizará el sistema
- »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”

25

# Ingeniería de Requerimientos

---

- » “Ingeniería de requerimientos 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”

26

# 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

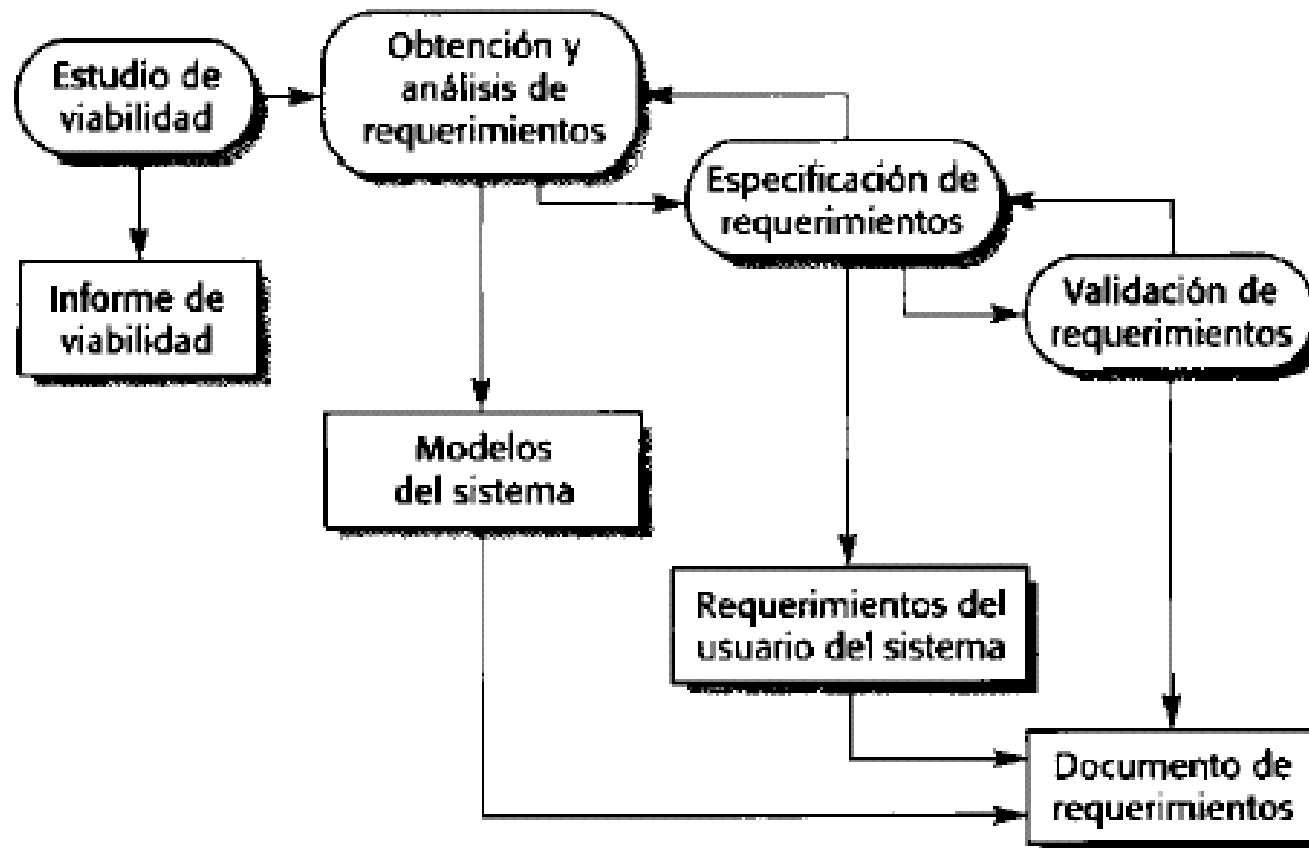
Mejora la comunicación entre equipos

Evita rechazos de usuarios finales.

27

# Ingeniería de Requerimientos

## »Proceso



28

# Ingeniería de Requerimientos

---

- »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?
- »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.

29

# Ingeniería 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.

30

# Ingeniería de Requerimientos

## Obtención y análisis de requerimientos

---

### »Tipos de requerimientos

#### Requerimientos funcionales

*Describen una interacción entre el sistema y su ambiente. Como 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.*

31

# Ingeniería de Requerimientos

## Obtención y análisis de requerimientos

---

### »Tipos de requerimientos

#### 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.

32



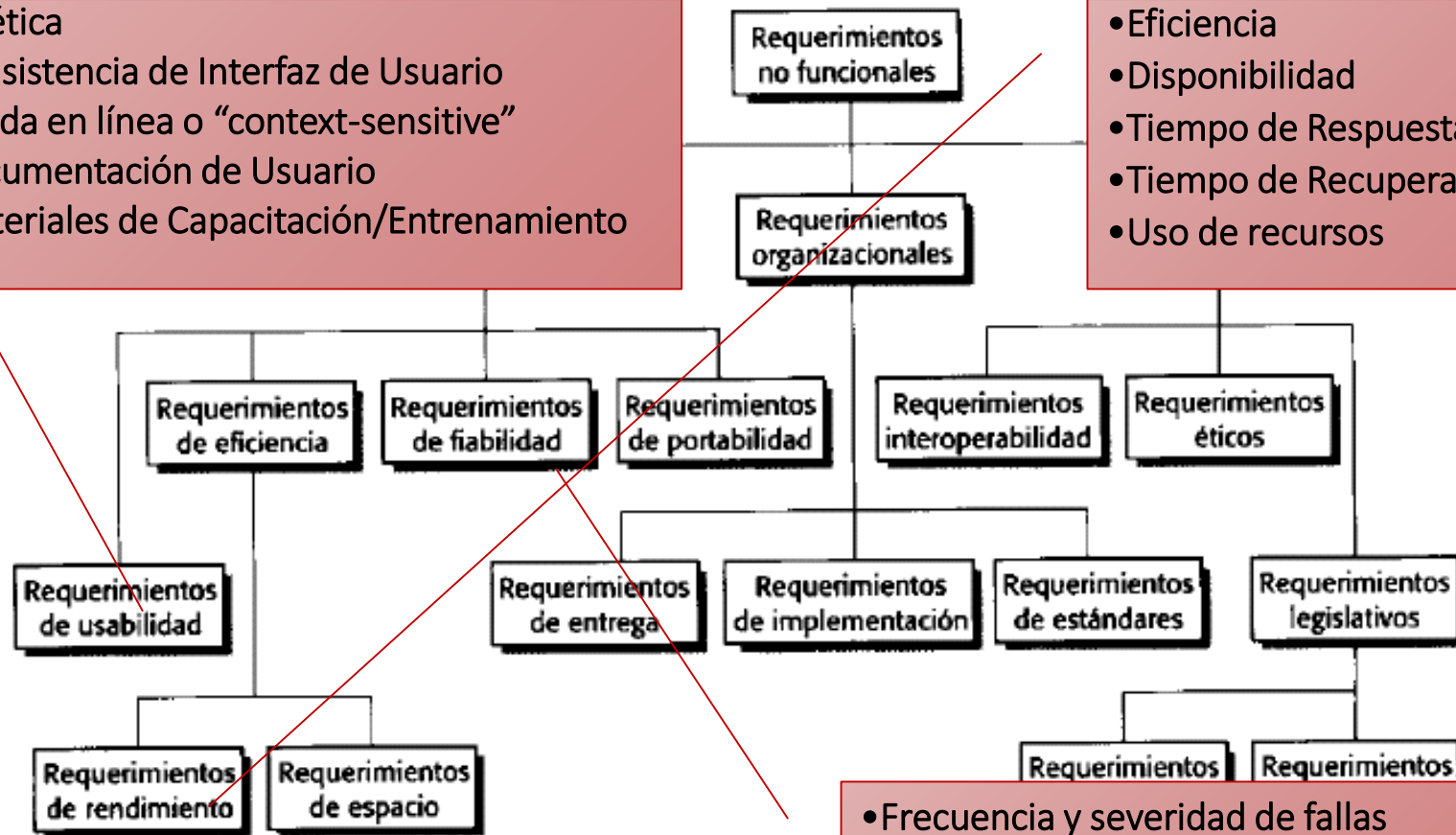
# Ingeniería de Requerimientos

## Obtención y análisis de requerimientos

»Tipos de r  
Requerim

- Estética
- Consistencia de Interfaz de Usuario
- Ayuda en línea o “context-sensitive”
- Documentación de Usuario
- Materiales de Capacitación/Entrenamiento

- Eficiencia
- Disponibilidad
- Tiempo de Respuesta
- Tiempo de Recuperación
- Uso de recursos



- Frecuencia y severidad de fallas
- Facilidades de recuperación
- Posibilidades de predicción

33

# Ingeniería de Requerimientos

## Obtención y análisis de requerimientos

---

### »Tipos de requerimientos

#### Otras Clasificaciones

##### *Requerimientos del dominio*

Reflejan las características y restricciones del dominio de la aplicación del sistema. Pueden ser funcionales o no funcionales y pueden restringir a los anteriores. Como se especializan en el dominio son complicados de interpretar.

##### *Requerimientos por Prioridad*

Que deben ser absolutamente satisfechos

Que son deseables pero no indispensables

Que son posibles, pero que podrían eliminarse

34

# Ingeniería de Requerimientos

## Obtención y análisis de requerimientos

---

### »Tipos de requerimientos

#### Otras Clasificaciones

##### *Requerimientos del Usuario*

Son declaraciones en lenguaje natural y en diagramas de los servicios que se espera que el sistema provea y de las restricciones bajo las cuales debe operar.

Pueden surgir problemas por falta de claridad, confusión de requerimientos, conjunción de requerimientos.

##### *Requerimientos del Sistema*

Establecen con detalle los servicios y restricciones del sistema.

Es difícil excluir toda la información de diseño (arquitectura inicial, interoperabilidad con sistemas existentes, etc.)

35

# Ingeniería de Requerimientos

---

## »Objetivos

Permiten que los desarrolladores expliquen cómo han entendido lo que el cliente pretende del sistema

Indican a los diseñadores qué funcionalidad y características va a tener el sistema resultante

Indican 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.

36

# Ingeniería de Requerimientos

## Especificación de requerimientos

---

### »Propiedades de la 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

37

# Ingeniería de Requerimientos

## 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.*

38

# Ingeniería 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.*

39

# Ingeniería 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

40



# Ingeniería de Requerimientos

## Validación de requerimientos

---

### »Definición de la IEEE

#### Validación

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

#### Verificación

*Determinar si un producto de software de una fase cumple los requerimientos de la fase anterior*

### »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

41

# Ingeniería de Requerimientos

## 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.

42

# Ingeniería de Requerimientos

## 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 pueden diseñar conjunto de pruebas)

43

# Ingeniería de Requerimientos

## Validación de requerimientos

---

### » Técnicas de validación

Pueden ser manuales o automatizadas

Revisiones de requerimientos (formales o informales)

Construcción de prototipos

Generación de casos de prueba

44

# Ingeniería de Requerimientos

## Validación de requerimientos

---

### »Revisión de Requerimientos

Es un proceso manual que involucra a distintas personas.

Ellos verifican el documento de requerimientos en cuanto a anomalías y omisiones.

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.

45

# Requerimientos

---

»Verificamos y validamos los requerimientos

¿Los requerimientos son correctos? (Correctitud)

¿Los requerimientos son consistentes? (Consistencia)

¿Los requerimientos son completos? (Compleitud)

¿Los requerimientos son realistas? (Verificabilidad)

¿Describe cada requerimiento algo que es necesario para el cliente?

¿Las personas que adquieren el sistema comprenden correctamente el requerimiento ?  
(Comprensibilidad)

¿Son adaptables los requerimientos ? (Adaptabilidad)

¿Esta claramente establecido el origen del requerimiento? (Trazabilidad)

46



# Gestión de los Requerimientos

---

» ¿Por que cambian los requerimientos?

Porque al analizar el problema, no se hacen las preguntas correctas a las personas correctas (En sistemas grandes hay una comunidad diversa de usuarios)

Porque los clientes y los usuarios son distintos

Porque cambió el problema que se estaba resolviendo

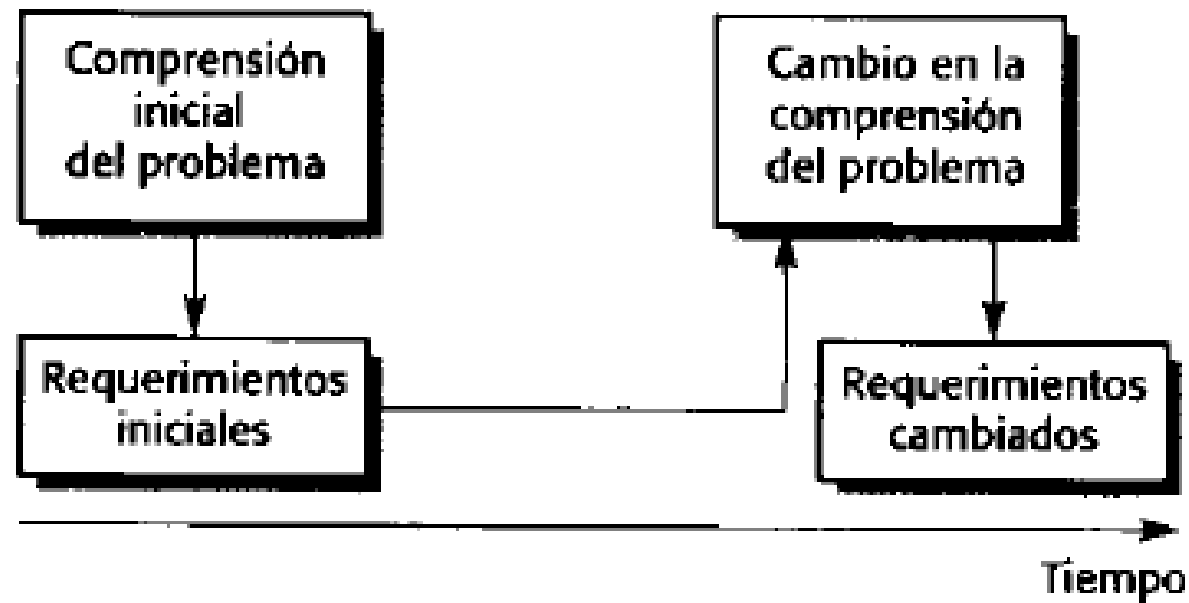
Porque los usuarios cambiaron su forma de pensar o sus percepciones

Porque cambió el ambiente de negocios (mercado, etc.)

47

# Gestión de los Requerimientos

## »Procesos



48



# Gestión de los Requerimientos

---

## »Evolución

### Requerimientos duraderos

*Relativamente estables, se derivan de la actividad principal de la organización.*

### Requerimientos volátiles

*Cambian durante el desarrollo del sistema o después que se puso en operación. (Ej: cambios gubernamentales)*

49

# Gestión de los Requerimientos

---

## »Requerimientos volátiles:

Requerimientos Cambiantes: cambian porque se modifica el ambiente (entorno)

Requerimientos emergentes: surgen como ampliación (al incrementar la comprensión del cliente)

Requerimientos consecuentes: surgen por la introducción del sistema. Pueden cambiar los procesos de la organización por desarrollar nuevas formas de trabajo.

Requerimientos de compatibilidad: cambian porque interactúan con otros sistemas que cambian

50

# Gestión de los Requerimientos

---

»Se debe realizar :

1. Identificación de requerimientos
2. Gestión del cambio
  - Análisis del problema y especificación del cambio
  - Análisis del cambio y cálculo de costos
  - Implementación del cambio
3. Políticas de rastreo
  - Fuente - Requerimientos - Diseño
4. Ayuda de herramientas CASE
  - Almacenar requerimientos
  - Gestionar el cambio
  - Gestionar el rastreo

51

# Técnicas de Especificación de Requerimientos

52

# 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 como 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.

*Referencia indirecta*

*Relaciones de recurrencia*

*Definición axiomática*

*Expresiones regulares*

*Abstracciones de datos*

*Otras...*

53

# 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.

*Tablas de decisión*

*Diagramas de transición de estados*

*Tablas de transición de estados*

*Diagramas de persianas*

*Diagramas de transición extendidos,*

*Redes de Petri*

*Otras...*

54

# Técnicas de Especificación de Requerimientos Estáticas

---

## »Referencia indirecta (ecuaciones implícitas)

Descripción del sistema con una referencia indirecta al problema y su solución.

Se define "QUÉ" se hace, no "CÓMO".

Ejemplo: sistema que resuelva k ecuaciones con n incógnitas => NO se declara el método de resolución, puede NO existir la solución.

## »Relaciones de recurrencia

Descripción del sistema mediante una función que define su valor en función de términos anteriores.

Ejemplo: Expresar la serie de Fibonacci

$$F(0) = 1 \quad F(1) = 1 \quad F(n+1) = F(n) + F(n-1)$$

55

# Técnicas de Especificación de Requerimientos Estáticas

---

## »Definición axiomática

Se definen las propiedades básica de un sistema a través de operadores y axiomas (debe ser un conjunto completo y consistente)

Se generan teoremas a través del comportamiento del sistema y se demuestran

Ejemplos: Sistemas expertos, Definición de TADs, etc.

56



# Técnicas de Especificación de Requerimientos Estáticas

## »Expresiones regulares

Se define un alfabeto y las combinaciones permitidas. Cuando un sistema procesa un conjunto de cadenas de datos, permite definir las cadenas de datos aceptables

### *Alfabeto*

ÁTOMOS: (símbolos básicos)  $a, b, c$ .

ALTERNACIÓN:  $(a|b) = \{a, b\}$

COMPOSICIÓN:  $(ab) = \{ab\}$

ITERACIÓN:  $(a)^* = \{e, a, aa, \dots\}$   $(a)^+ = \{a, aa, \dots\}$

### *Se definen las combinaciones válidas*

$(a(b|c)) = \{ab, ac\}$

$(a(b|c))^+ = \{ab, ac, abac, acab, \dots\}$

57

# Técnicas de Especificación de Requerimientos Estáticas

---

## » Abstracciones de datos

Para aquellos sistemas en los que los datos determinan las clases de acciones que se realizan (importa para qué son).

Se categorizan los datos y se agrupan los semejantes.

El diccionario contiene los TIPOS DE DATOS (clases) y los DATOS (objetos).

Se organizan de tal manera de aprovechar las características compartidas.

»

58

# Técnicas de Especificación de Requerimientos Dinámicas

---

## »Tablas de Decisión

Es una herramienta que permite presentar de forma concisa las reglas lógicas que hay que utilizar para decidir acciones a ejecutar en función de las condiciones y la lógica de decisión de un problema específico.

## »Describe el sistema como un conjunto de:

Posibles CONDICIONES satisfechas por el sistema en un momento dado

REGLAS para reaccionar ante los estímulos que ocurren cuando se reúnen determinados conjuntos de condiciones y

ACCIONES a ser tomadas como un resultado.

59

# Técnicas de Especificación de Requerimientos Dinámicas

## »Tablas de Decisión

Construiremos las tablas con:

*condiciones simples y acciones simples*

*Las condiciones toman sólo valores Verdadero o Falso*

*Hay  $2N$  Reglas donde  $N$  es el nro. de condiciones*

	REGLA 1	REGLA 2	.....
COND 1			
COND 2			
.....			
ACCION 1			
ACCION 2			
.....			

60

# Técnicas de Especificación de Requerimientos Dinámicas

---

## »Tablas de Decisión

Modelizar el problema de remisión de mercadería con las siguientes consideraciones:

- 1- *Si el comprador no es cliente se imprime un mensaje de aviso y no se remite.*
- 2- *Si no hay stock y el comprador es cliente no se remite.*
- 3- *Si hay stock y el comprador es cliente se remite*

61

# Técnicas de Especificación de Requerimientos Dinámicas

---

## »Tablas de Decisión

Modelizar el problema de remisión de mercadería con las siguientes consideraciones:

1. Si el *comprador* no *es cliente* se imprime un mensaje de aviso y no se remite.
2. Si no *hay stock* y el *comprador es cliente* no se remite.
3. Si *hay stock* y el *comprador es cliente* se remite

62

# Técnicas de Especificación de Requerimientos Dinámicas

---

## »Tablas de Decisión

Modelizar el problema de remisión de mercadería con las siguientes consideraciones:

1. Si el *comprador* no *es cliente* se *imprime un mensaje de aviso* y *no se remite*.
2. Si no *hay stock* y el *comprador es cliente* no se remite.
3. Si *hay stock* y el *comprador es cliente* se remite

63

# Técnicas de Especificación de Requerimientos Dinámicos

1. Si el *comprador* no *es cliente* se *imprime un mensaje de aviso* y *no se remite*.
2. Si no *hay stock* y el *comprador es cliente* no se remite.
3. Si *hay stock* y el *comprador es cliente* se remite

Condiciones

Acciones

	Reglas			
Es cliente	V	V	F	F
Hay stock	V	F	V	F
Imprime mensaje de aviso			X	X
Se remite	X			
No se remite		X	X	X

64



# Técnicas de Especificación de Requerimientos Dinámicas

---

## » Tablas de Decisión

### Especificaciones **completas**

*Aquellas que determinan acciones (una o varias) para todas las reglas posibles.*

### Especificaciones **redundantes**

*Aquellas que marcan para reglas que determinan las mismas condiciones acciones iguales.*

### Especificaciones **contradictorias**

*Aquellas que especifican para reglas que determinan las mismas condiciones acciones distintas.*

65

# Técnicas de Especificación de Requerimientos Dinámicas

## »Tablas de Decisión

### Redundancia y Contradictoria

	Reglas						
C1	V	V	.	..	..	F	F
C2	V	V	.	.	.	V	V
C3	V	F	.	.	..	F	F
A1			.	.	..	X	X
A2	X				.		
A3		X	.	.		X	X

Redundancia

	Reglas						
C1	V	V	.	.	..	F	F
C2	V	V	.	.	.	V	V
C3	V	F	.	.	..	F	F
A1			.	.	..		X
A2	X				.	X	
A3		X	.	.		X	

Contradictoria

66

# Técnicas de Especificación de Requerimientos Dinámicas

## »Tablas de Decisión

### Reducción de Complejidad (Redundancia)

*Combine las reglas en donde sea evidente que una alternativa no representa una diferencia en el resultado.*

*La raya [—] significa que la condición 2 puede ser S o N, y que aún así se realizará la acción.*

Condición 1:	S	S
Condición 2	S	N
Acción 1	X	X

Condición 1:	S
Condición 2	—
Acción 1	X

67

# Técnicas de Especificación de Requerimientos Dinámicas

## »Tablas de Decisión

Reducción de Complejidad (Redundancia)

*Algebra de bool*

	Reglas			
<b>Es cliente</b>	V	V	F	F
<b>Hay stock</b>	V	F	V	F
<b>Imprime mensaje de aviso</b>			X	X
<b>Se remite</b>	X			
<b>No se remite</b>		X	X	X

Reglas		
V	V	F
V	F	—
		X
X		
	X	X

68

# Técnicas de Especificación de Requerimientos Dinámicas

---

»En el sistema de un video club se quiere modelizar el subsistema de alquileres. Solo se alquilan películas si la copia de la película está disponible. Un socio puede tener en su poder 3 películas. Si ya las tiene en el momento de alquilar otra película, no se permite alquilarla. Si tiene películas vencidas sin devolver, no se le permite alquilar y se le cancela momentáneamente la cuenta. Si no está la copia de la película, no tiene 3 películas en su poder y no tiene alquileres vencidos se le reserva la película

69

# Técnicas de Especificación de Requerimientos Dinámicas

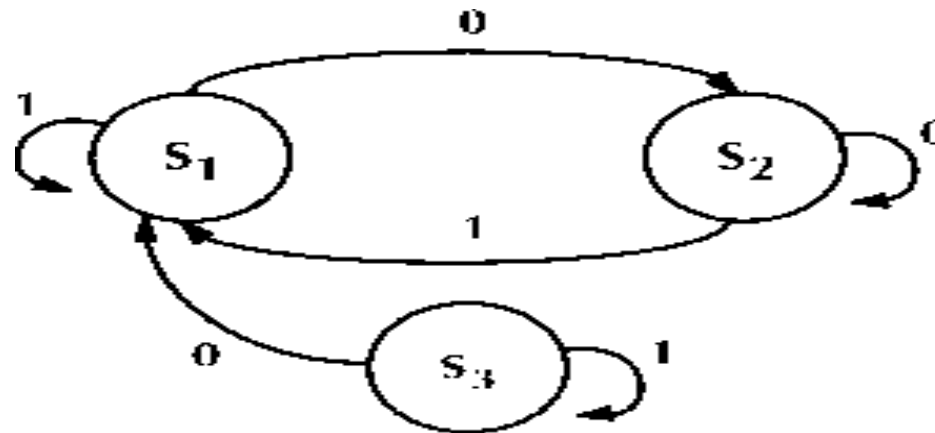
## » Maquinas de Estado Finito

Describe al sistema como un conjunto de estados donde el sistema reacciona a ciertos eventos posibles (externos o internos).

»  $f(S_i, C_j) = S_k$

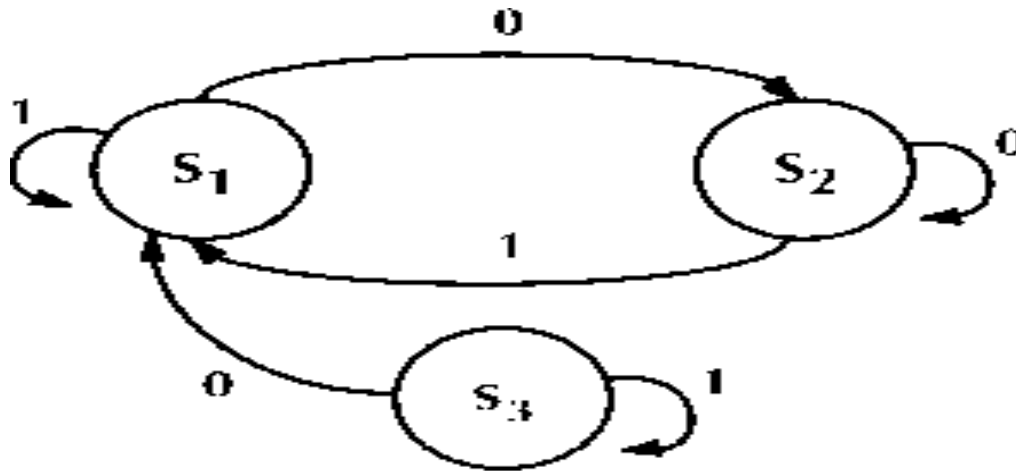
Al estar en el estado  $S_i$ , la ocurrencia de la condición  $C_j$  hace que el sistema cambie al estado  $S_k$ .

70



# Técnicas de Especificación de Requerimientos Dinámicos

## »Maquinas de Estado Finito



$$f(S1, 0) = S2$$

$$f(S1, 1) = S1$$

$$f(S2, 0) = S2$$

$$f(S2, 1) = S1$$

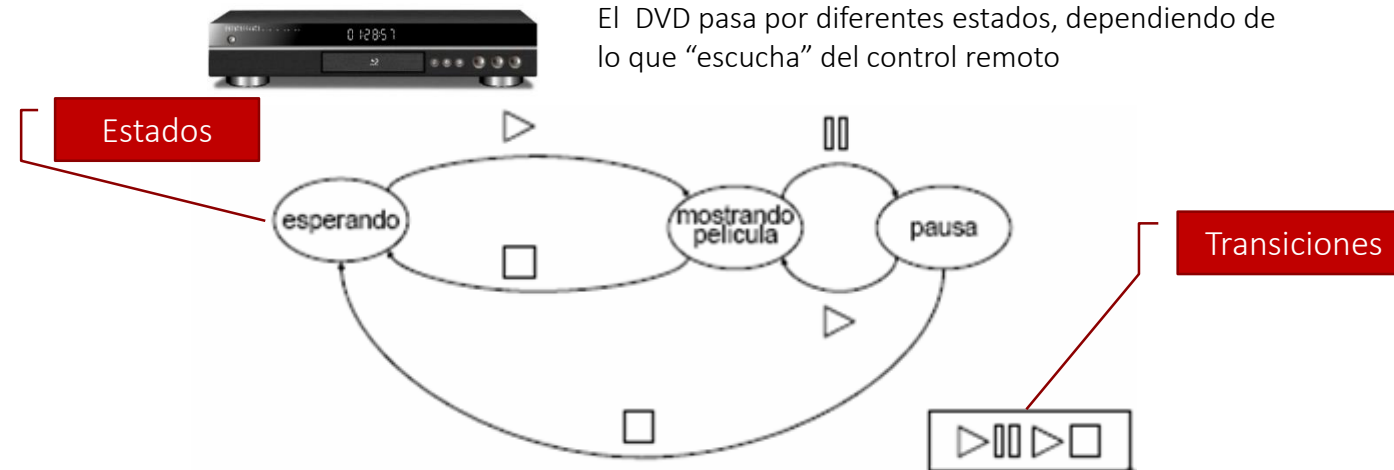
$$f(S3, 0) = S1$$

$$f(S3, 1) = S3$$

71

# Técnicas de Especificación de Requerimientos Dinámicos

## » Máquinas de Estado Finito

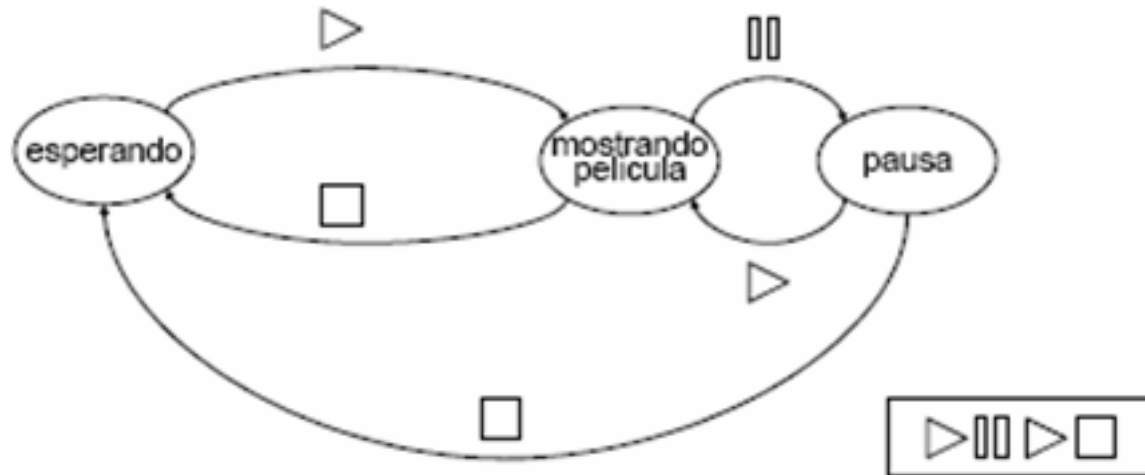


72



# Técnicas de Especificación de Requerimientos Dinámicos

## »Maquinas de Estado Finito



$$f(A, \triangleright) = B$$

$$f(B, \square) = A$$

$$f(B, \text{||}) = C$$

$$f(C, \triangleright) = B$$

$$f(C, \square) = A$$

73

# Técnicas de Especificación de Requerimientos Dinámicos

---

## »Maquinas de Estado Finito

### Definición formal

*Formalmente, un autómata finito (AF) puede ser descrito como una 5-tupla  $(S, \Sigma, T, s, A)$  donde:*

*$\Sigma$  es un alfabeto;*

*$S$  un conjunto de estados;*

*$T$  es la función de transición;*

*$s$  es el estado inicial;*

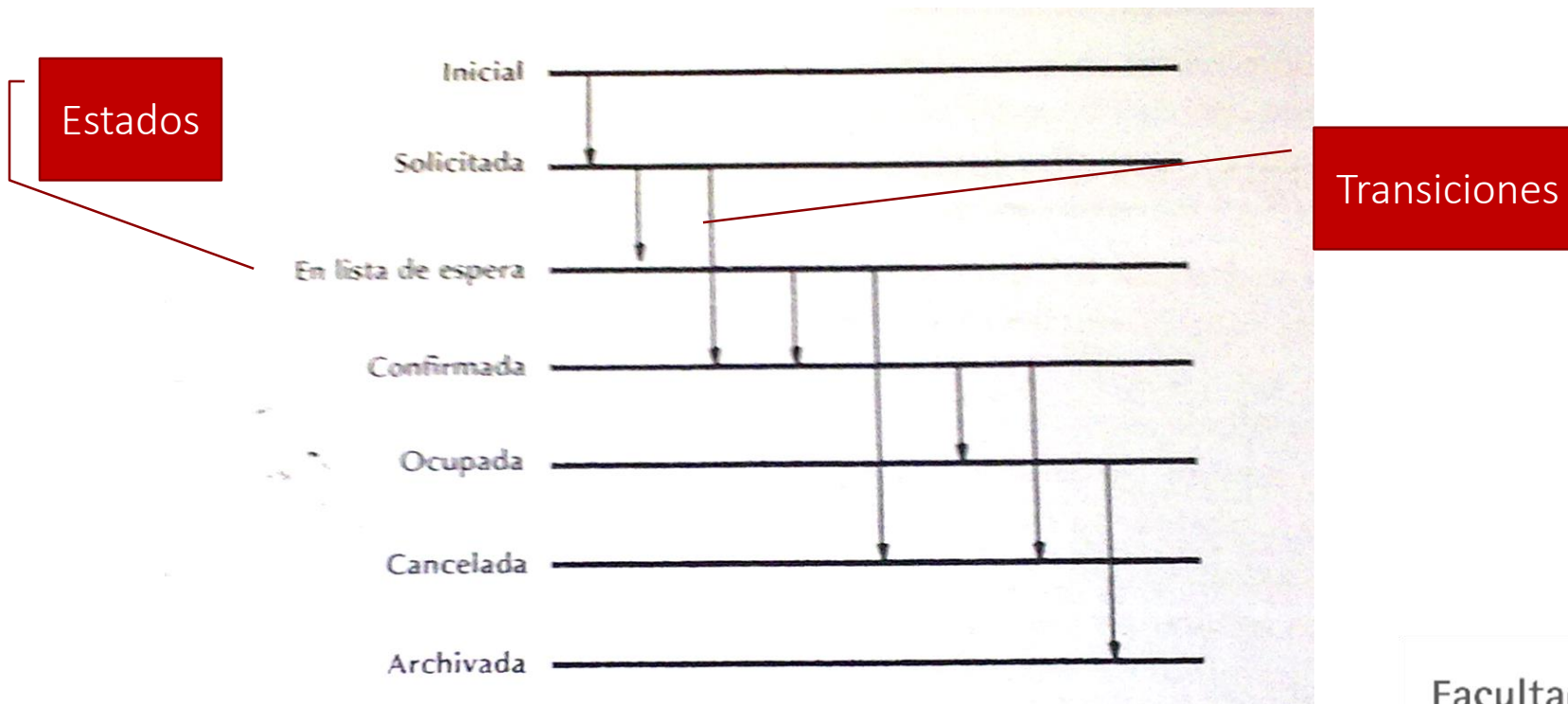
*$A$  es un conjunto de estados de aceptación o finales.*

74

# Técnicas de Especificación de Requerimientos Dinámicos

## »Maquinas de Estado Finito

Representación en grafico de persiana



75

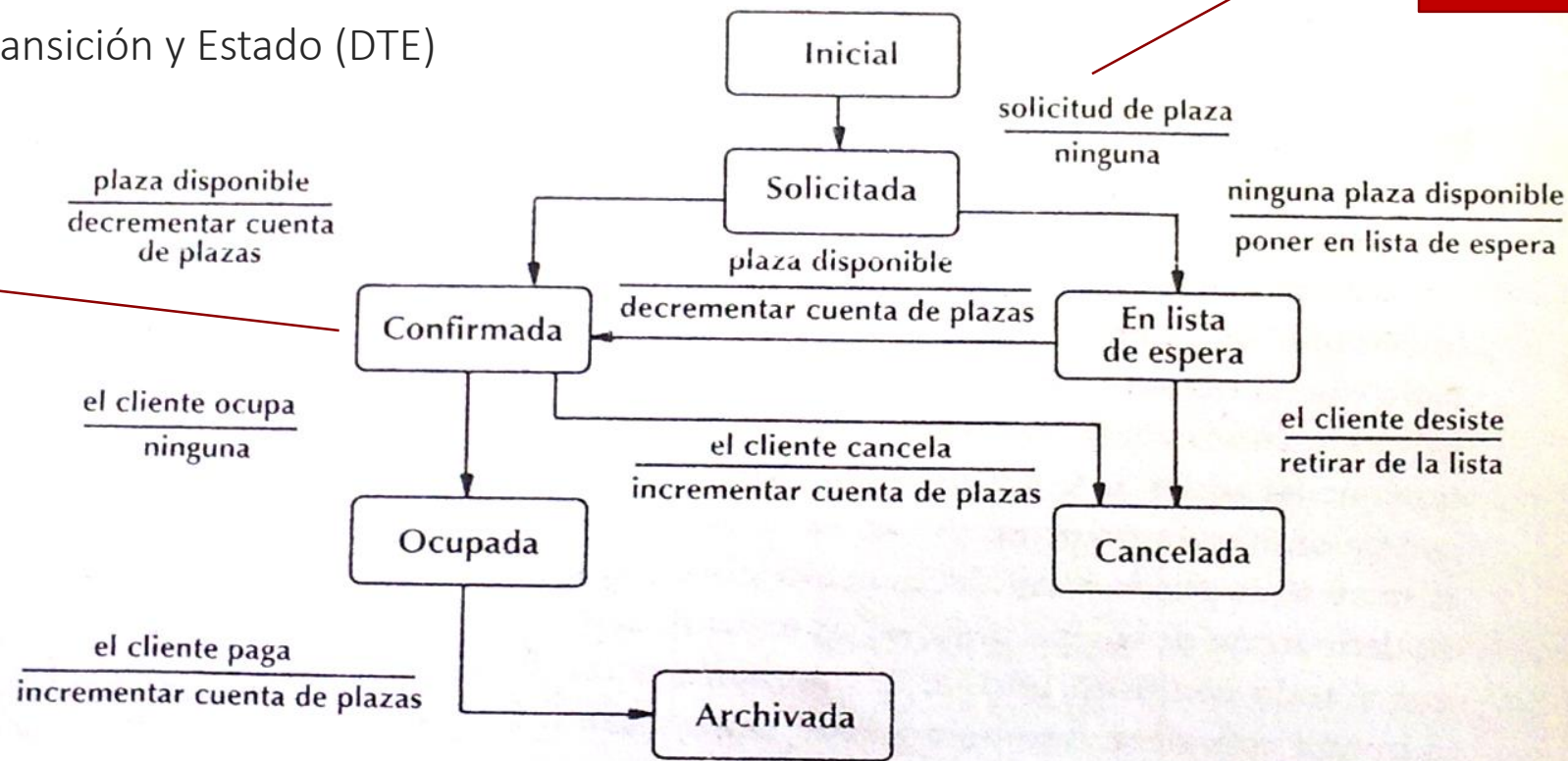
# Técnicas de Especificación de Requerimientos Dinámicos

## »Maquinas de Estado Finito

Diagrama de Transición y Estado (DTE)

Transiciones

Estados



76

# Técnicas de Especificación de Requerimientos Dinámicas

---

## » Construcción de un DTE

- 1- Identificar los estados
- 2- Si hay un estado complejo se puede explotar
- 3- Desde el estado inicial, se identifican los cambios de estado con flechas
- 4- Se analizan las condiciones y las acciones para pasar de un estado a otro
- 5- Se verifica la consistencia:

*Se han definido todos los estados*

*Se pueden alcanzar todos los estados*

*Se pueden salir de todos los estados*

*En cada estado, el sistema responde a todas las condiciones posibles (normales y anormales)*

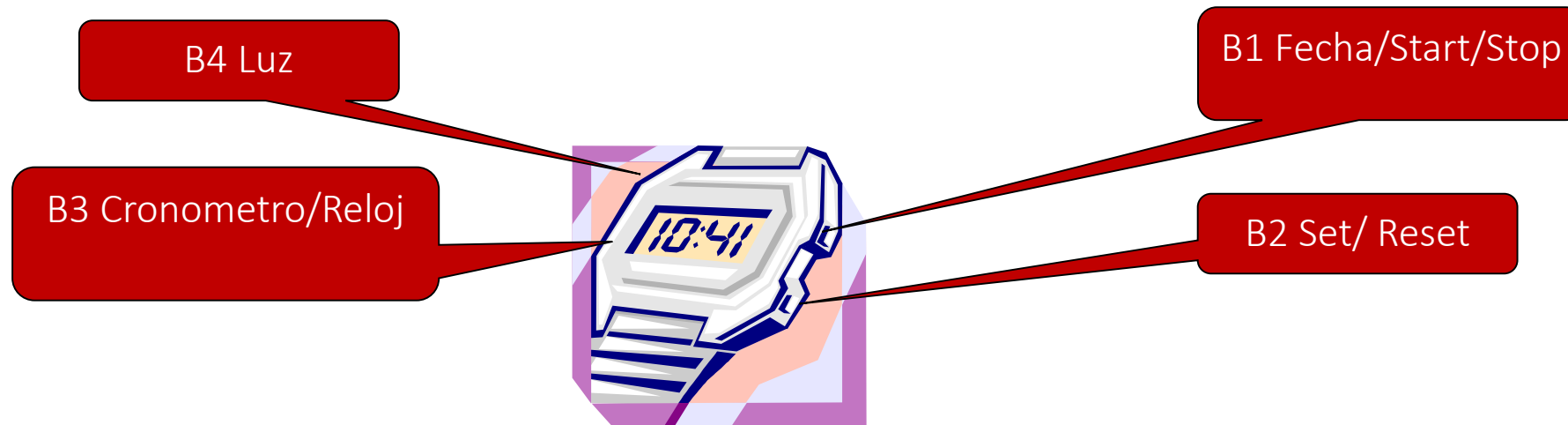
77

# Técnicas de Especificación de Requerimientos

## Ejercicio DTE

### »Reloj Cronometro

El reloj posee una pantalla y 4 botones



78

# Técnicas de Especificación de Requerimientos

## Ejercicio DTE

---

### »Funciones

Inicialmente (al colocar la pila) visualiza la hora prefijada

Visualizar la hora

Visualizar la fecha

Modificar Hora y Fecha

Encender la Luz por 5 seg.

Iniciar / Detener / Resetear Cronometro

Deja de funcionar al finalizarse la pila

79

# Técnicas de Especificación de Requerimientos

## Ejercicio DTE

### »1- Identificar los estados

Visualizando hora

Visualizando fecha

Visualizando funciones cronometro

Cronometrando

Configurando hora y fecha

### »2- Identificar estados complejos

No es necesario

### »3- Estado inicial

En este caso, el sistema inicia al colocarse la pila y pasaría al estado visualizando hora





# Técnicas de Especificación de Requerimientos

## Ejercicio DTE

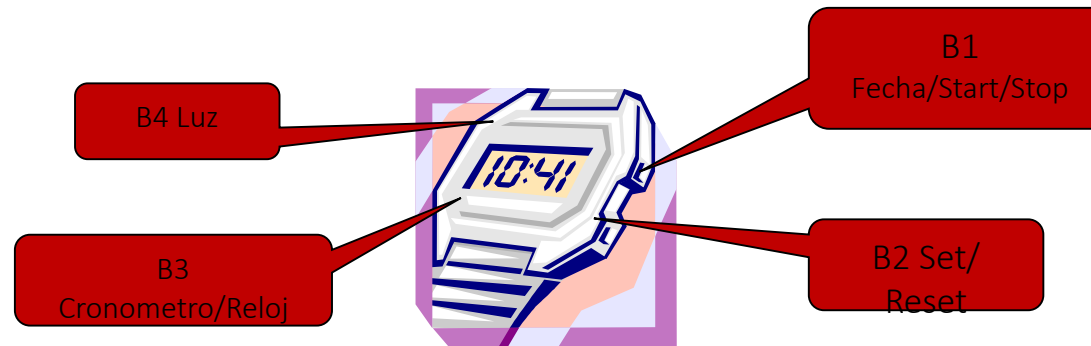
### »4- Visualizando hora

Se presiona B1 Visualiza la fecha

Se presiona B2 Modificar la hora y fecha

Se presiona B3 Visualiza el cronometro

Se presiona B4 Enciende la luz



81

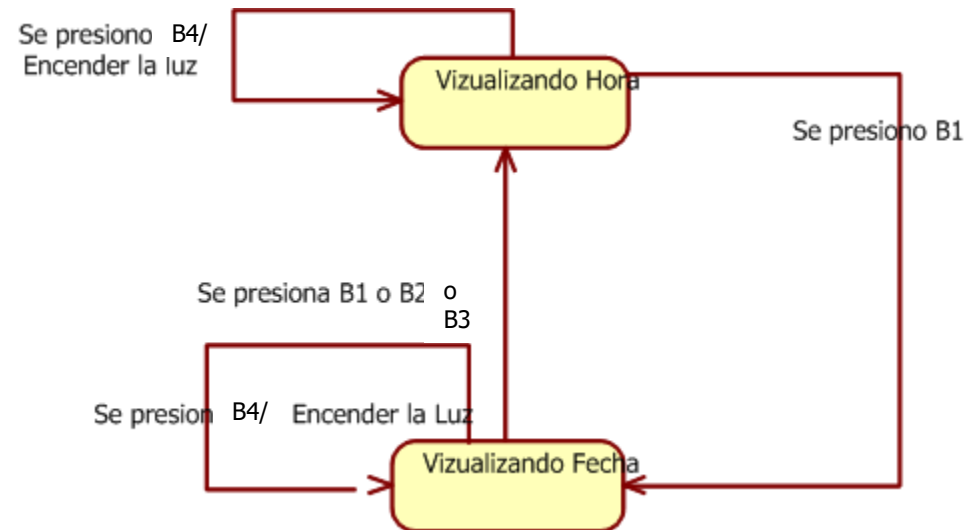
# Técnicas de Especificación de Requerimientos

## Ejercicio DTE

### »4- Visualizando fecha

Para visualizar la fecha se debe presionar el botón B1 y luego presionando B1 o B2 o B3 vuelve a visualizar la hora

En Cualquier Momento se puede encender la luz con el botón B4



82

# Técnicas de Especificación de Requerimientos

## Ejercicio DTE

---

### »4- Configurando Hora y Fecha

Se presiona B1 modifíco el dígito

Se presiona B2 vuelve a visualizar la hora

Se presiona B3 Modifíco el dígito a modificar

*Hora, minuto, segundo, día, mes*

Se presiona B4 enciende la luz

### »4- Continuar con todos los estados

83

# Técnicas de Especificación de Requerimientos

## Ejercicio DTE

---

»5- Se verifica la consistencia:

Se han definido todos los estados

Se pueden alcanzar todos los estados

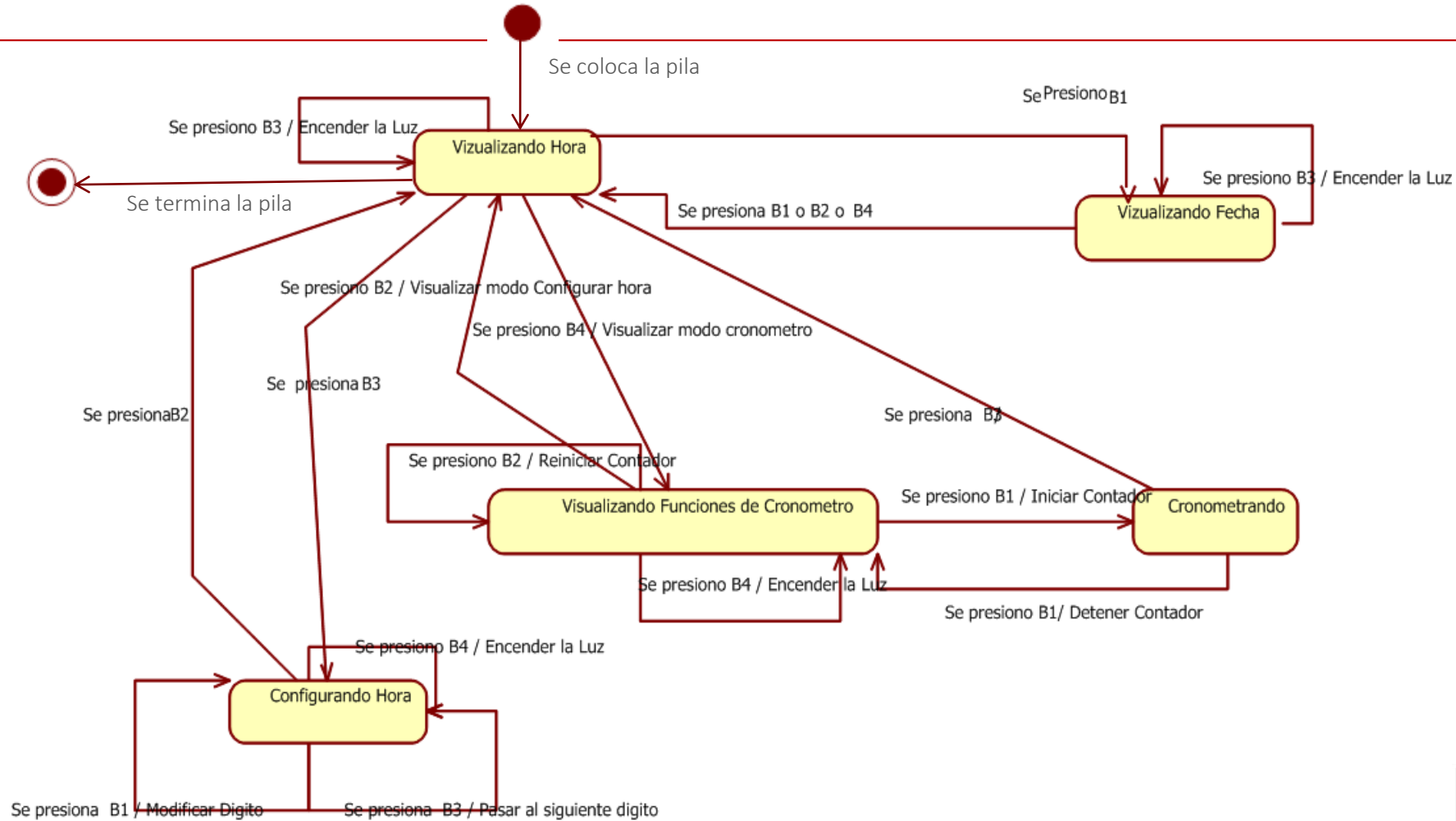
Se pueden salir de todos los estados

En cada estado, el sistema responde a todas las condiciones posibles (normales y anormales)

84

# Técnicas de Especificación de Requerimientos

## Ejercicio DTE



85