CAPITULO I : INTRODUCCIÃ N A LOS S.I.

• EL CONCEPTO DE SISTEMA

- Definiciones.
- Elementos.
- Relaciones.
- Enfoque SistÃ©mico.

DEFINICIONES

- "Sistema" : Concepto o herramienta genÃ©rica para analizar el funcionamiento de un Ã¡rea.

- "Sistema" (segÃºn el DRAE) : Conjunto de cosas utilizadas para lograr un objetivo.

ELEMENTOS DEL SISTEMA

- Componentes del sistema.
- Relaciones entre elementos. (Que determinan la estructura del sistema)
- Objetivo del sistema.

OTROS ELEMENTOS

- Entorno del sistema.
- LÃ–mites del sitema. ("Frontera entre el sistema y el entorno")

REALCIONES DEL SISTEMA

- Entre componentes: permiten considerarlo como 1 unidad. Aseguran su cohesiÃ³n.
- Con el exterior. ("Entradas y salidas del sistema)

ENFOQUE SISTÃ MICO U HOLÃ STICO

- Manera de estudiar o analizar sistemas mediante una visiÃ³n global y refinamientos sucesivos (descomposiciÃ³n de arriba-abajo: Top-down).
- 1. Se toma todo el sistema como una caja negra de la que sÃ³lo nos interesan sus entradas y salidas.
- 2. Se identifican los subsistemas y las relaciones entre ellos.
- 3. Se llevan a cabo descomposiciones sucesivas hasta que los componentes sean tan simples que se puedan estudiar fÃ¡cilmente
- Muy Ãºtil en aplicaciones de software grande.

• EL CONCEPTO DE INFORMACIÃ N

- Diferencia entre dato e informaciÃ³n
- Calidad de InformaciÃ³n
- Propiedades de la calidad de informaciÃ³n

DIFERENCIA ENTRE DATO E INFORMACIÃ N

- Datos: se constituyen por registros de los hechos, acontecimientos, transacciones…

- InformaciÃ³n: “datos procesados”. Son Ãºtiles o significativos para el receptor.

- Se consideran los datos como la “materia prima para obtener informaciÃ³n”.
- Procesar = “indicar el significado de los datos”.
- Procesar = “dar significaciÃ³n a un dato en un contexto”.

CALIDAD DE INFORMACIÃ N

- Conjunto de cualidades que disminuyen la incertidumbre y ayudan al receptor a tomar la decisiÃ³n mÃ¡s ventajosa.

PROPIEDADES DE LA CALIDAD DE INFORMACIÃ N

- Relevante para el el propÃ³sito de la decisiÃ³n o el problema considerado
- Precisa (exacta con la realidad). Que se pueda confiar en ella. No existe la `precisÃ³n absoluta'.
- Suficientemente completa para el problema. Lo “ideal” es contar con toda la informaciÃ³n relevante. Lo mÃ¡s importante es que la informaciÃ³n sobre los elementos clave sea completa.
- Se comunica a la persona adecuada para la decisiÃ³n.
- Se comunica a tiempo para que pueda ser Ãºtil.
- Llega al nivel de detalle mÃ¡s adecuado.
- Es comprensible para el receptor.

• LOS SISTEMAS DE INFORMACIÃ N

- Funciones de la infraestructura de una empresa
- Distintas estructuras en las empresas (mariconadas)
- Definiciones de Sistema de InformaciÃ³n
- Elementos de un S.I.
- Estructura de un S.I.
- Flujos de informaciÃ³n en una empresa
- TecnologÃ–as de la InformaciÃ³n
- Otros conceptos

FUNCIONES DE LA INFRAESTRUCTURA DE UNA EMPRESA

- Controlar y Gestionar recursos (funciÃ³n o sistema contable y de gestiÃ³n econÃ³mica).
- Comercializar (Actividad comercial y de ventas)
- Fabricar servicios o crear productos (ProducciÃ³n)

DISTINTAS ESTRUCTURAS EN LAS EMPRESAS (Mariconadas)

- Por funciones y departamentos.
    - ♦ Control y GestiÃ³n.
    - ♦ Comercio.
    - ♦ ProducciÃ³n.
- Por localizaciÃ³n geogrÃ¡fica.
- Por jerarquÃ–a y niveles de actuaciÃ³n.

DEFINICIONES DE SI

Las definiciones mÃ¡s empleadas de S.I. parten de la base de que lo principal es el objetivo perseguido.

- **Def 01** : Conjunto formal de procesos (operando sobre una colecciÃ³n de datos estructurada segÃºn las necesidades de la empresa) que recopilan, elaboran y distribuyen la informaciÃ³n necesaria para llevar a cabo las operaciones de empresa y las actividades de direcciÃ³n y control.
- **Def 02 (by Inma)** : El sistema encargado de coordinar los flujos y los registros de informaciÃ³n necesarios para desarrollar una actividad de acuerdo a su estrategia o planteamiento.

ELEMENTOS DE UN S.I.

- Procedimientos y prÃ¡cticas habituales de trabajo.
- InformaciÃ³n.
- Personas o usuarios.
- Equipo de soporte (parte fÃ–sica: papel, mÃ¡quina de escribir, ordenadores…)

- La informaciÃ³n de un S.I. puede ser **informal** o **formal**:
  - **InformaciÃ³n formal**: la informaciÃ³n "fÃ–sica". De la que tenemos constancia y podemos confiar en ella.
  - **InformaciÃ³n informal**: (Ej.) rumores, diÃ¡logos personales…

I

ESTRUCTURA DE UN S.I.

- **Operaciones y transacciones**: se encarga del procesamiento de actividades diarias o transacciones (acontecimientos rutinarios: pagos, facturaciÃ³n…). Las transacciones constituyen la mayor parte de las actividades que se realizan en la empresa.
  - CaracterÃ–sticas de las transacciones:
    - Los procedimientos de tratamiento se comprenden bien y se pueden describir en detalle.
    - Hay pocas excepciones a procedimientos normales.
    - Gran volumen de transacciones.
    - Gran similitud entre transacciones.

    - Los procedimientos de transacciones han de describir, tanto los pasos a seguir en una situacion normal, como en una excepcional.

- **Nivel Operativo de DirecciÃ³n**: analizar los resultados (usualmente respecto a los recursos: diner, tiempo…) consumidos en las transacciones para tomar decisiones a corto plazo. Normalmente suele trabajar con informaciÃ³n del registro de transacciones.
  - CaracterÃ–sticas de la informaciÃ³n empleada:
    - Repetitiva (informes periÃ³dicos de ventas, pagos…)
    - Centrada en el pasado (resultados histÃ³ricos)
    - Con datos originados internamente
    - Datos con formato bien estructurado
    - Datos detallados y precisos.

- **Nivel tÃ¡ctico de direcciÃ³n**: asignaciÃ³n efectiva de recursos a medio plazo (se suele actuar con un aÃ±o de antelaciÃ³n), para mejorar el rendimiento de la empresa.
  - Tipos de informes anailizados
    - ResÃºmenes con medias estadÃ–sticas.
    - De excepciones (departamentos que han consumido mÃ¡s de la media, centros con pÃ©rdidas…)
    - EspecÃ–ficos (informes que no se han pedido antes y que los directivos necesitan

urgentemente para resolver problemas muy concretos)

- Se trabaja con el propÃ³sito de descubrir algo nuevo en los datos.

- **Nivel EstratÃ©gico de DirecciÃ³n**: pretende decidir las pautas a seguir por la empresa en el futro (plazos largos: 3-5 aÃ±os). La informaciÃ³n que maneja ha de tener un formato muy resumido para lograr predecir lo mejor para el Ã©xito futuro de la compaÃ±Ã–a. Normalmente se toman decisiones con un fuerte componente subjetivo.

FLUJOS DE INFORMACIÃ N EN LA EMPRESA

- **Flujos vericales ascendentes**: (subordinado superior). Informes sobre resultados de actividades. CarÃ¡cter histÃ³rico y de orientaciÃ³n al pasado.
- **Flujos verticales descendentes**: (jefe subordinado). Ã rdenes y solicitudes especÃ–ficas.
- **Flujos horizontales**: (entre personas del mismo nivel). Al contenido de esta informaciÃ³n se le llama "informaciÃ³n de coordinaciÃ³n". Se consideran estos flujos un medio esencial para adaptarse mejor al mercado.

TECNOLOGÃ AS DE LA INFORMACIÃ N

- **Def:** Sofisticadas herramientas que nos permiten implantar los actuales sistemas de informaciÃ³n.

OTROS CONCEPTOS

- **(MIS) Sistema de informaciÃ³n para la GestiÃ³n**: proporcionan a los directivos informaciÃ³n y ayuda para decisiones `estructuradas'. Es la parte del S.I. dediacada al nivel operativo tÃ¡ctico y estratÃ©gico.
- **(DSS) Sistema de apoyo a las decisiones**: similar al MIS. Da soporte a decisiones poco estructuradas.
- **Sistema de procesamiento de transacciones**: para el tratamiento de operaciones rutinarias diarias o transacciones.

"POSIBLES" PREGUNTAS CAPÃ TULO I

1. SeÃ±ala la respuesta **incorrecta**, Â¿El enfoque TOP-DOWN?:

- ♦ Se utiliza para el desarrollo de aplicaciones software.
- ♦ Consiste en analizar y desarrollar un sistema por refinamiento sucesivo mediante descomposiciÃ³n de arriba abajo.
- ♦ Se denomina tambiÃ©n Botton-Up y consiste en dividir el sistema en subsistemas considerÃ¡ndolos cajas negras de las que lo que mÃ¡s nos interesa es su interior.
- ♦ Es una filosofÃ–a de anÃ¡lisis y diseÃ±o que permite descomponer un trabajo complicado en tareas de programaciÃ³n mÃ¡s sencillas.

2. Â¿CuÃ¡l de las siguientes caracterÃ–sticas **no** forma parte de la calidad de informaciÃ³n?:

- ♦ Se aconseja que sea comunicada a tiempo, pero no es primordial.
- ♦ Debe tener relevancia y complitud.
- ♦ Debe llegar a la persona adecuada y con el nivel de detalle mÃ¡s adecuado, para que esta persona pueda decidir con dicha informaciÃ³n.
- ♦ Debe ser fiable y comprensible.

- Â¿CuÃ¡l de las siguientes afirmaciones **es correcta**? Los flujos de informaciÃ³n en la empresa:

    - Los que circulan desde el personal de base a los directivos se llaman verticales descendentes.
    - Los que van desde superiores al personal subordinado se llaman horizontales ascendentes.
    - Los que no aparecen en los organigramas de las empresas se llaman comunicaciones informales.

- Â¿CuÃ¡l de las siguientes afirmaciones **no** es correcta?:

    - Un S.I. forma parte de todos los subsistemas que puedan detectarse en una empresa.
    - No pueden existir sistemas de soporte de informaciÃ³n puramente manuales.
    - Un S.I. No tiene por quÃ© estar totalmente automatizado.
    - Cabe distinguir entre S.I., S.I.A. y Sistema InformÃ¡tico de Soporte.

- Â¿CuÃ¡l de las siguientes afirmaciones **es correcta**?:

    - El Ãºnico objetivo de una empresa es ganar dinero.
    - Un dato siempre estÃ¡ guardado en una variable.
    - Un dato es informaciÃ³n si disminuye nuestro nivel de incertidumbre.
    - La informaciÃ³n debe ser tratada siempre informÃ¡ticamente.

En exÃ¡menes anteriores…

- Los principales elementos que encontramos en cualquier sistema:

    - Componentes, relaciones que determinan la estructura y objetivos.
    - Objetivos, lÃ–mites y estructura.
    - Componentes, relaciones y entorno.

- Indica cuÃ¡les son los elementos de un S.I.:

    - Los procedimientos y prÃ¡cticas de trabajo, los objetivos y las relaciones.
    - Los procedimientos y prÃ¡cticas de trabajo, la informaciÃ³n, las personas y el equipo de soporte.
    - Los objetivos, la informaciÃ³n y los procedimientos y prÃ¡cticas de trabajo.

- Â¿CuÃ¡l de las siguientes afirmaciones **no** es correcta?:

    - Las prÃ¡cticas de trabajo determinan la informaciÃ³n que se necesita en un S.I.
    - Los objetivos de la empresa determinan las prÃ¡cticas de trabajo.
    - La informaciÃ³n determina las carÃ¡cterÃ–sticas del equipo Humano del S.I.

- Â¿CuÃ¡l de las siguientes afirmaciones **es correcta**?:

    - La informaciÃ³n debe ser tratada siempre informÃ¡ticamente.
    - Un dato es informaciÃ³n si disminuye nuestro nivel de incertidumbre.
    - El Ãºnico objetivo de una empresa es ganar dinero.

- La tÃ©cnica de anÃ¡lisis de sistemas:

    - Estudia todas sus entradas.
    - Sigue un enfoque sistÃ©mico.

- Estudia el sistema adoptando una visiÃ³n parcial del mismo hasta llegar a su visiÃ³n global.

• La estructura de un S.I.:

- Se puede describir mediante una pirÃ¡mide en la que se distingue la jerarquÃ–a de diversos niveles de actuaciÃ³n y gestiÃ³n.
- Se puede describir mediante un grafo de niveles segÃºn una organizaciÃ³n geogrÃ¡fica.
- Se puede describir mediante un organigrama segÃºn departamentos o funciones.

CAPITULO II : SISTEMAS DE INFORMACIÃ N BÃ"SICOS EN LAS EMPRESAS

SUBSISTEMA DE RECUROS HUMANOS	
ACTIVIDADES	• Plantilla • NÃ³mina
NIVEL OPERATIVO	• Mantenimiento de datos de los empleados • Inventario de cualificaciones de empleados • Inventario de puestos de trabajo y condiciones • EvaluaciÃ³n de empleados • GeneraciÃ³n de informes • GestiÃ³n de solicitudes de empleo • EnvÃ–o de instrucciones de pago de salarios
NIVEL TÃ"CTICO	• Perfil de la persona ideal PRO puesto de trabajo • Necesidades de contrataciÃ³n de personal • Generar planes PRO incentivos y beneficios a empleados • AnÃ¡lisis de necesidades de informaciÃ³n y creaciÃ³n de planes para mejorar el nivel tÃ¡ctico-profesional de la plantilla
NIVEL ESTRATÃ GICO	• Crear planes para mejorar la infraestructura de la empresa
TECNOLOGÃ AS DE LA INFORMACIÃ N	• RealizaciÃ³n de nÃ³minas mediante aplicaciones de trabajo por lotes • GestiÃ³n de personal mediante tratamientos interactivos • ProtecciÃ³n de datos mediante control de acceso
SUBSISTEMA DE CONTABILIDAD Y FINANZAS	
ACTIVIDADES	• Llevar la contabilidad
NIVEL OPERATIVO	• Control de activos fijos • GestiÃ³n de cobros • GestiÃ³n de pagos • Control de inventario • EjecuciÃ³n de nÃ³mina • GeneraciÃ³n de informes
NIVEL TÃ"CTICO	• GestiÃ³n y control de presupuestos • InformaciÃ³n sobre el flujo de caja • Controles de planes de gasto de capital

NIVEL ESTRATÃ GICO	• Trabajo de forma interactiva • Trabajo con gran volumen de datos		
TECNOLOGÃ AS DE LA INFORMACIÃ N	• AutomatizaciÃ³n de transacciones • GestiÃ³n econÃ³mica mediante (grandes) bases de datos • AnÃ¡lisis financieros mediante simulaciones muy elaboradas • AnÃ¡lisis estadÃ−stico		
SUBSISTEMA COMERCIAL Y DE VENTAS			
ACTIVIDADES	**Ventas**	• GestiÃ³n y tratamiento de pedidos • FacturaciÃ³n • Control de detalles de entrega y actualizaciÃ³n de inventario	
		ComercializaciÃ³n	• InformaciÃ³n de ventas • InvestigaciÃ³n de mercados • Informes tÃ©cnicos de producciÃ³n • Datos sobre la capacidad financiera de la empresa
	NIVEL OPERATIVO	**Apoyo a vendedores**	• GestiÃ³n de carteras de clientes • Control de contactos con clientes • Consultas sobre productos • InformaciÃ³n sobre crÃ©dito o consideraciÃ³n econÃ³mica de cliente • Facilidades para la gestiÃ³n de pedidos y facturas
		GestiÃ³n de distribuciÃ³n de productos	• Control de envÃ−os • RecepciÃ³n • Devoluciones
NIVEL TÃ”CTICO	• Recogida de informaciÃ³n de ventas • GestiÃ³n y control del marketing • Establecimiento de precios • Decidir la mejor forma de distribuciÃ³n • AnÃ¡lisis de competidores		
NIVEL ESTRATÃ GICO	• Dividir mercado		

	• Seleccionar segmentos a acceder • Planificar productos y servicios a ofertar • Predecir ventas a trabajar
TECNOLOGÃ AS DE LA INFORMACIÃ N	• Pedidos y facturaciÃ³n mediante aplicaciones de trabajo por lotes • GestiÃ³n comercial mediante grandes bases de datos • AnÃ¡lisis de mercado mediante simulaciones complejas y anÃ¡lisis estadÃ–sticos

SUBSISTEMA DE PRODUCCIÃ N	
ACTIVIDADES	• Control de existencias almacenadas
NIVEL OPERATIVO	• Compras de materias primas o componentes • RecepciÃ³n de materias primas o componentes • EnvÃ–o de productos a clientes
NIVEL TÃ"CTICO	• GestiÃ³n y control de materias primas y productos • PlanificaciÃ³n de la capacidad de producciÃ³n Ã³ptima
NIVEL ESTRATÃ GICO	• (.) Suelen ser decisiones estratÃ©gicas de alta direcciÃ³n
TECNOLOGÃ AS DE LA INFORMACIÃ N	• Sistemas de control y automatizaciÃ³n de producciÃ³n

RESPUESTAS

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
c	a	c	b	c	a	b	c	b	b	a

CAPITULO III : CICLO DE VIDA DEL SOFTWARE

- Concepto de Ciclo de Vida.
- Procesos del Ciclo de Vida.
- Modelo en cascada.
- Modelo incremental.
- Modelo en espiral.

• CONCEPTO DE CICLO DE VIDA

- Los departamentos de un S.I. requieren un marco de referencia…
  - ♦ …que pueda ser empleado por todas las personas que participan en un desarrollo informÃ¡tico.
  - ♦ …en que se definan los procesos, las actividades y las tareas a desarrollar.

DEFINICIONES

- **SegÃºn IEE 1074**: "El Ciclo de Vida es una aproximaciÃ³n lÃ³gica a la adquisiciÃ³n, el suministro, el desarrollo, la explotaciÃ³n y el mantenimiento del software".

- **SegÃºn ISO 12207-1**: “El ciclo de vida es un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotaciÃ³n y el mantenimiento de un producto software, abarcando la vida del sistema desde la definiciÃ³n de requisitos hasta la finalizaciÃ³n de su uso”.

- El Ciclo de Vida abarca **toda** la vida del sistema (desde su “concepciÃ³n” hasta que ya “no se utiliza”).

• PROCESOS DEL CICLO DE VIDA DEL SOFTWARE

- Procesos Principales.
- Procesos de Soporte.
- Procesos Generales.
- Proceso de AdaptaciÃ³n.

• MODELO EN CASCADA

- CaracterÃ–sticas.
- CrÃ–ticas.

CARACTERÃ STICAS

- Cada fase empieza cuando ha terminado la anterior.
- Para pasar de una fase a otra se han de conseguir todos los objetivos de la fase previa.
- Ayuda a prevenir que se sobrepasen las fechas de entrega y los costes esperados.
- Al final de cada fase, el personal tÃ©cnico y los usuarios pueden pueden revisar el progreso del proyecto.

- Es el Ciclo de Vida mÃ¡s antiguo y el mÃ¡s utilizado.

CRÃ TICAS

- No refleja el “proceso real” del desarrollo del software NO contempla iteraciones en etapas no contiguas.
- Se requiere mucho tiempo para pasar todo el ciclo (hasta que no se finalice una etapa no se comienza la siguiente).
- AcentÃºa el fracaso de la industria del software con el usuario final El sistema no funciona hasta casi el final.

• MODELO INCREMENTAL

CARACTERÃ STICAS

- Corrige la necesidad de una secuencia NO lineal aÃ±adiendo componentes funcionales al sistema (“incrementos”).
- En cada paso se actualiza el sistema.
- El sistema software acaba siendo la integraciÃ³n de los resultados sucesivos (obtenidos despuÃ©s de cada iteraciÃ³n).

- Se ajusta a entornos de alta incertidumbre.

CRÃ TICAS

- Cuenta con el problema de que es difÃ–cil determinar si los requisitos propuestos son vÃ¡lidos Los errores en los requisitos son detectados tarde y su correcciÃ³n es costosa.

- **MODELO EN ESPIRAL**

- CaracterÃ–sticas.
- Diferencias respecto a los modelos tradicionales.
- Dificultades.
-

CARACTERÃ STICAS

- Consta de una serie de **ciclos**. Para cada uno de los ciclos, primero se identifican sus objetivos, alternativas y restricciones. Posteriormente se lleva a cabo dicho ciclo para, una vez acabado Ã©ste, comenzar a plantear el siguiente.
- Cada ciclo se completa con una revisiÃ³n.

DIFERENCIAS RESPECTO A LOS MODELOS TRADICIONALES

- Reconocimiento explÃ–cito de diferentes alternativas para lograr los objetivos del proyecto.
- IdentificaciÃ³n de riesgos asociados a las alterntivas y la manera de resolverlos (“centro del modelo”).
- DivisiÃ³n de proyectos en ciclos.
- El modelo se adapta a cualquier tipo de actividad.

DIFICULTADES

- Trabaja bien en desarrollos internos, pero necesita ajustes posteriores para adaptarlo a la subcontrataciÃ³n de software.
- Necesidad de expertos en evaluaciÃ³n de riesgos para identificar y manejar los fuertes riesgos del proyecto.

“POSIBLES” PREGUNTAS CAPÃ TULO III

1. Indicar la(s) respuesta(s) correcta(s). El Ciclo de Vida:

- ♦ Comienza con una idea o necesidad que satisfacer y acaba con las pruebas satisfactorias del producto.
- ♦ No existe ningÃºn estÃ¡ndar que describa sus procesos y actividades.
- ♦ No es sÃ³lo realizar el anÃ¡lisis, diseÃ±o, codificaciÃ³n y pruebas; tambiÃ©n incluye entre otros, procesos de soporte.
- ♦ El Mantenimiento son las actividades para mantener sin cambios el sistema.
- ♦ En la actividad de AnÃ¡lsis de los Requisitos del Software, los desarrolladores obtienen de los futuros usuarios los requisitos que piden al sistema.

En exÃ¡menes anteriores…

- El Ciclo de Vida del software:

- ♦ Es un marco de referncia comÃºn para las personas que participan en el desarrollo

informÃ¡tico.
- ♦ Se definen sÃ³lo los mÃ©todos a desarrollar.
- ♦ Empieza en el anÃ¡lisis y termina cuando ya no se utiliza el software.

- Los procesos del Ciclo de Vida:

  - ♦ La norma de ISO al respecto agrupa los procesos en generales, de desarrollo y de la organizaciÃ³n.
  - ♦ La norma de ISO al respecto agrupa los procesos principales en adquisiciÃ³n, suministro, desarrollo, explotaciÃ³n y mantenimiento.
  - ♦ La norma de ISO al respecto agrupa los procesos generales en adquisiciÃ³n, suministro, desarrollo, explotaciÃ³n y mantenimiento.

- En cuanto a los modelos del Ciclo de Vida:

  - Las normas no fomentan ni especifican ningÃºn modelo en concreto.
  - En el modelo en Cascada pueden desarrollarse fases en paralelo, de ahÃ– su nombre.
  - El modelo incremental exige la necesidad de contar con expertos en evaluaciÃ³n de riesgos.

- Los siguientes factores influyen a la hora de elegir un ciclo de vida:

  - La estreucturaciÃ³n del sistema, la familiaridad con la tecnologÃ–a a usar.
  - La norma estÃ¡ndar que se use en el desarrollo.
  - Las prisas que tenga el cliente por tener en marcha la aplicaciÃ³n.

- El Ciclo de Vida:

  - Empieza con una idea o necesidad de satisfacer y acaba con las pruebas satisfactorias del producto.
  - El mantenimiento lo constituyen las actividades para mantener sin cambios el S.I.
  - En la actividad de AnÃ¡lisis de los Requisitos Software, los desarrolladores obtienen de los futuros usuarios los requisitos que piden al sistema.

RESPUESTAS

1	2	3	4	5	6
c - e	a	b	a	a	c

CapÃ–tulo III: Ciclo de Vida del Software

Este ejercicio lo pasÃ³ Inma a principios de curso (puede que fuese la tercera semana, no tengo ganas ahora de comprobarlo). El caso es que consistÃ–a en, una vez leÃ–da la descripciÃ³n de una actividad, indicar la fase o etapa del ciclo de vida con el que se correspondÃ–a.

26 cuestiones de las que sÃ³lo “acertÃ©” 6 en su momento.

En su momento. TambiÃ©n. Detalle bastante importante: estuvimos comprobando con los apuntes todo el tiempo, lo cual induce a pensar que tampoco deberÃ–a aparecer una cuestiÃ³n de este tipo en el examen. De todas formas, para seguir con la buena costumbre de no obviar nada, como este ejercicio `relativamente teÃ³rico' apareciÃ³n en clase, he aquÃ– su correspondiente rÃ©plica en el Breathe Proyect 0.2 .

E intentando no desviarnos, hacer menciÃ³n a que varias de las “descripciones” Inma las tratÃ³ como “definiciones propiamente dichas de las fases”. Ã stas se indican convenientemente.

Por fin, sin mÃ¡s dilaciones:

Identificar la fase o etapa del ciclo de vida:

- Organizar el equipo de personas que va a llevar a cabo el estudio de un SI, asÃ– como la elaboraciÃ³n de un calendario de ejecuciÃ³n.
- Proponer una soluciÃ³n a corto plazo que tenga en cuenta las restricciones econÃ³micas, tÃ©cnicas, legales y operativas.
- Determinar quÃ© usuarios participan en la obtenciÃ³n de requisitos.
- Elaborar un modelo de datos.
- Identificar servidores, monitores, impresoras que formarÃ¡n parte del equipo de un SI.
- Especificar infraestructura tecnolÃ³gica necesaria para soportar un SI.
- Determinar la formaciÃ³n necesaria para el personal encargado de la implantaciÃ³n de un SI.
- Comprobar el funcionamiento correcto de un SI en su entrono operativo.
- Estudiar quÃ© parte de un SI se ve afectada cuando realizamos una modificaciÃ³n concreta.
- Elaborar un modelo de SI vÃ¡lido que soporte a los procesos de la organizaciÃ³n.
- Recogida de requisitos que debe cumplir un software para identificar funcionalidades.
- Identificar objetivos estratÃ©gicos a los que apoya un SI, asÃ– como el Ã¡mbito general de la organizaciÃ³n a la que afectarÃ¡.
- Especificar los niveles de servicio que se prestarÃ¡n una vez implantado el SI.
- Recogida y anÃ¡lisis de los antecedentes generales que puedan afectar a los procesos y unidades organizativas implicadas en un SI.
- Proponer diversas alternativas que respondan a los requisitos planteados para un SI.
- Posibilidad de efectuar un pago tanto en metÃ¡lico como en efectivo.
- Especificar los requisitos de implantaciÃ³n de un SI relacionados con la formaciÃ³n, infraestructura e instalaciÃ³n.
- Estudio del SI actual.
- Identificar los factores crÃ–ticos para el Ã©xito de un SI, asÃ– como sus participantes nombrando mÃ¡ximos responsables.
- Generar un diagnÃ³stico estimando la eficiencia de los existentes SI identificando posibles problemas y mejoras.
- Definir los formatos de pantalla, diÃ¡logos e informes.
- DiseÃ±ar un modelo fÃ–sico de datos.
- Codificar los componentes de un SI a partir de las especificaciones del diseÃ±o.
- Verificar los recursos necesarios disponibles para realizar la instalaciÃ³n de todos los componentes de un SI.
- Definir quÃ© compromisos se adquieren con la entrega del SI.
- CreaciÃ³n y puesta a punto de Base de Dato.

SOLUCIONES	
1	Proceso de AdquisiciÃ³n.
2	Proceso de Suministro.
3	Proceso de Desarrollo - AnÃ¡lisis de Requisitos del Sistema.
4	Proceso de Desarrollo - AnÃ¡lisis de Requisitos del Software.
5	Proceso de Desarrollo - DiseÃ±o de la Aquitectura del Sistema.
6	Proceso de Desarrollo - DiseÃ±o de la Arquitectura del Software.
7	Proceso de ExplotaciÃ³n.
8	Proceso de ExplotaciÃ³n.
9	Proceso de Mantenimiento.
10	Proceso de AdquisiciÃ³n.

11	Proceso de Desarrollo - AnÃ¡lisis de Requisitos del Sistema.
12	Proceso de AdquisiciÃ³n.
13	Proceso de Suministro.
14	Proceso de AdquisiciÃ³n.
15	Proceso de Suministro.
16	Proceso de Desarrollo - AnÃ¡lisis de Requisitos del Sistema.
17	Proceso de Desarrollo - DiseÃ±o de la Arquitectura del Software.
18	Proceso de AdquisiciÃ³n **(DefiniciÃ³n)**.
19	Proceso de AdquisiciÃ³n.
20	Proceso de Suministro **(DefiniciÃ³n)**.
21	Proceso de Desarrollo - AnÃ¡lisis de Requisitos del Software.
22	Proceso de Desarrollo - DiseÃ±o Detallado del Software.
23	Proceso de Desarrollo - CodificaciÃ³n y Prueba del Sofware.
24	Proceso de ExplotaciÃ³n.
25	Proceso de Suministro.
26	Proceso de ExplotaciÃ³n.

CAPITULO IV : METODOLOGÃ AS DE DESARROLLO DE SOFTWARE

- Conceptos generales.
- EvoluciÃ³n histÃ³rica.
- CaracterÃ–sticas.
- ClasificaciÃ³n.
- Principales metodologÃ–as.

• CONCEPTOS GENERALES

- Origen.
- Definiciones.
- Distinciones.
- Especificaciones.
- Necesidades a cubrir.
- Objetivos.
- Anexo 0.

APARICIÃ N DE LAS METODOLOGÃ AS

Las metodologÃ–as surgen de la necesidad de unas determinadas reglas fijas a las que ceÃ±irse a la hora de desarrollar un software.

DEFINICIONES

- “MetodologÃ–a”: conjunto de pasos y procedimientos para el desarrollo de software.
- “MetodologÃ–a”: camino para desarrollar software de manera sistemÃ¡tica.

DISTINCIONES

- **Ciclo de Vida**: indica **QUÃ** productos obtengo.
- **MetodologÃ–a**: indica **CÃ MO** obtengo los productos.

- **MÃ©todo**: se aplica a **cada fase** del ciclo de vida.
- **MetodologÃ–a**: conjunto de mÃ©todos que se aplican durante **todo el ciclo de vida**.

ESPECIFICACIONES DE UNA METODOLOGÃ A

Una metodologÃ–a ha de indicar:

- CÃ³mo se debe **dividir** un proyecto en etapas.
- QuÃ© **tareas** se llevan a cabo en cada etapa.
- QuÃ© **salidas** se producen y cuÃ¡ndo se deben producir.
- QuÃ© **restricciones** se aplican.
- QuÃ© **herramientas** se van a utilizar.
- CÃ³mo se **gestiona y controla** un proyecto.

NECESIDADES A CUBRIR CON UNA METODOLOGÃ A

- **Mejores aplicaciones**: (considerando mejores sistemas los de mejor calidad).

- **Mejor proceso de desarrollo**: identificar las salidas de las fases para planificar y controlar el proyecto. AsÃ– los sistemas se desarrollan mÃ¡s rÃ¡pidamente y con los recursos apropiados.

- **Proceso estÃ¡ndar en la organizaciÃ³n**: (aporta claros beneficios).

OBJETIVOS DE LAS METODOLOGÃ AS

- Registrar los requisitos del S.I de forma adecuada.
- Proporcionar un mÃ©todo sistemÃ¡tico de desarrollo para controlar su progreso.
- Construir un S.I. en el tiempo adecuado y con unos costes aceptables.
- Construir un sistema bien documentado y fÃ¡cil de mantener.
- Ayudar a identificar lo mÃ¡s pronto posible cualquier cambio posible en el proceso de desarrollo.
- Proporcionar un sistema que satisfaga a todas las personas afectadas por el mismo.

PROCEDIMIENTOS, TÃ CNICAS Y HERRAMIENTAS

- **Procedimiento**: “define la forma de ejecutar una tarea. El resultado de aplicar un procedimiento es un producto”.
- **TÃ©cnica**: “se usa para aplcar un procedimiento”.
- **Herramienta**: “se usa para aplicar una tÃ©cnica”.

- **EVOLUCIÃ N HISTÃ RICA**

- Convencional.
- Estructurado.
- Orientado a Objetos.

ENFOQUE-DESARROLLO CONVENCIONAL

- CaracterÃ–sticas.
- Problemas.

- **CaracterÃ–ticas:**

- ♦ Sin metodologÃ–as de desarrollo.
- ♦ Distintos papeles: **analistas**, **programadores** (funcionales y orgÃ¡nicos) y **operadores**.

- **Problemas:**

  - ♦ Los resultados finales son impredecibles (no se sabe cuÃ¡ndo puede acabar realmente el proyecto).
  - ♦ No hay forma de controlar lo que estÃ¡ sucediendo en el proyecto (por no existir fases establecidas y productos intermedios concretos).
  - ♦ Los cambios organizativos afectan negativamente al proceso de desarrollo (por no existir documentos extandarizados o porque no se actualizan adecuadamente).

ENFOQUE-DESARROLLO ESTRUCTURADO

- Conceptos.
- ProgramaciÃ³n estructurada.
- DiseÃ±o estructurado.
- AnÃ¡lisis estructurado.

- **Conceptos:**

  - ♦ Se puede considerar el nacimiento de las *tÃ©cnicas estructuradas* como el origen de los desarrollos automatizados.
  - ♦ El origen del desarrollo estructurado comenzÃ³ con la programaciÃ³n estructurada.

- **ProgramaciÃ³n estructurada:**

  - ♦ Se pretendiÃ³ que la visiÃ³n de un programa fuese lo mÃ¡s *comprensible* posible.
  - ♦ *Normas* para la aplicaciÃ³n de las estructuras de datos y control.

- **DiseÃ±o estructurado:**

  - ♦ Se utiliza el *mÃ³dulo de programa* como componente bÃ¡sico de construcciÃ³n.
  - ♦ Se refina el concepto de *modularidad* (normalizando la estructura de un mÃ³dulo de programa, restringiendo las relaciones entre mÃ³dulos y estableciendo medidas en la calidad de los programas).

- **AnÃ¡lisis estructurado:**

  - ♦ Conceptos.
  - ♦ Diferencias respecto al anÃ¡lisis clÃ¡sico.
  - ♦ EvoluciÃ³n.

  - ♦ Conceptos:

    - ◊ Se conoce tambiÃ©n como “anÃ¡lisis descendente” o “top-down”.
    - ◊ Se utilizaba principalmente en organizaciones de desarrollo de sistemas de gestiÃ³n.

  - ♦ Diferencias respecto al anÃ¡lis clÃ¡sico:

Analisis ClÃ¡sico:

- Se hacÃ–a una especificaciÃ³n narrativa de los requisitos, tal como los percibÃ–a el analista.
- Las especificaciones contaban con el problema de ser *monolÃ–ticas*, *redundantes*, *ambiguas* e *imposibles de mantener*.

Analisis Estructurado:

- Movimiento gradual a las especificaciones funcionales.
- Las especifiaciones funcionales se caracterizaban por ser: *grÃ¡ficas*, *particionadas* y *mÃ–nimamente redundantes*.

♦ EvoluciÃ³n respecto al anÃ¡lisis estructurado clÃ¡sico:

◊ Dar menor importancia a la construcciÃ³n de los modelos fÃ–sicos actuales y modelos lÃ³gicos actuales.
◊ Diferenciar mÃ¡s los modelos fÃ–sicos y lÃ³gicos.
◊ Ampliar las tÃ©cnicas de anÃ¡lisis estructurado para poder modelar sistemas de tiempo real.
◊ Enfocarse en el modelo de datos.
◊ Estudio de los eventos.

ENFOQUE-DESARROLLO ORIENTADO AL OBJETO (No se estudia)

• CARACTERÃ STICAS PRINCIPALES DE LAS METODOLOGÃ A

- Impacto.
- CaracterÃ–sticas deseables.

INMPACTO DE LA METODOLOGÃ A EN EL ENTORNO DE DESARROLLO

- **La metodologÃ–a proporciona CALIDAD y PRODUCTIVIDAD en el desarrollo de software**.
- La metodologÃ–a estÃ¡ influenciada por consideraciones como el tamaÃ±o y la estructura de la organizaciÃ³n o el tipo de aplicaciones que se va a desarrollar.

CARACTERÃ STICAS DESEABLES DE UNA METODOLOGÃ A

- **Existencia de reglas predefinidas**: indicar reglas que definan las fases, tareas, productis intermedios, etc.

- **Cobertura total del ciclo de desarrollo**: indicar los pasos a seguir desde el planteamiento hasrta el mantenimiento (proporcionando mecanismos para integrar los resultados en la fase siguiente).

- **Verificaciones intermedias**: contemplar la realizaciÃ³n de verificaciones sobre los productos generados en cada fase para comprobar su correcciÃ³n.

- **PlanificaciÃ³n y control**: para proporcionar una forma de desarrollar software planificada y controlada, con objeto de que no se disparen costes ni se amplÃ–en los tiempos de entrega.

- **ComunicaciÃ³n efectiva**: para facilitar el trabajo en grupo y con los usuarios.

- **UtilizaciÃ³n sobre un abanico amplio de proyectos**: la metodologÃ–a ha de ser flexible para poder cubrir diverrsos proyectos.

- **FÃ¡cil formaciÃ³n**: los desarrolladores deben comprender las tÃ©cnicas y procedimientos de gestiÃ³n.

- **Herramientas CASE**: debe ser soportada por herramientas automatizadasa.

- **Contener actividades que mejoren el proceso de desarrollo**: se hace necesario conocer los resultados para hacer mediciones en cada etapa.

- **Soporte al mantenimiento**: para facilitar las modificaciones en sistemas existentes.

- **Soporte a la reutilizaciÃ³n del software**: incluir procedimientos para la creaciÃ³n, mantenimiento y recuperaciÃ³n de los componentes reutilizables (y no sÃ³lo del cÃ³digo).

• CLASIFICACIÃ N DE LAS METODOLOGÃ AS

- MetodologÃ–as Estructuradas.
- Sistemas de Tiempo Real.

METODOLOGÃ AS ESTRUCTURADAS

- Orientadas a procesos.
- Orientadas a datos.
- Mixtas.

- **MetodologÃ–as Orientadas a procesos:**

  - ♦ Conceptos.
  - ♦ Componentes de una especificaciÃ³n estructurada.
  - ♦ Principales metodologÃ–as.

  - ♦ Conceptos:

- EstÃ¡n basadas en un mÃ©todo descendente (Top-down) de descomposiciÃ³n funcional, para describir los requisitos del sistema.
- Se apoyan en *tÃ©cnicas grÃ¡ficas*, dando lugar a un nuevo concepto que es la *especificaciÃ³n estructurada*.

  - ♦ Componentes de una especificaciÃ³n estructurada:
  - ♦ Diagramas de Flujo de Datos (DFD).
  - ♦ Diccionario de Datos.
  - ♦ Especificaciones de proceso.

  - ♦ Principales metodologÃ–as estructuradas:
  - ♦ MetodologÃ–a de De Marco.
  - ♦ MetodologÃ–a de Gane y Sarson.
  - ♦ MetodologÃ–a de Yourdon/Constantine.

- **MetodologÃ–as Orientadas a datos:**

  - ♦ Orientadas a datos jerÃ¡rquicos.
  - ♦ Orientadas a datos no jerÃ¡rquicos.

  - ♦ MetodologÃ–as Orientadas a datos jerÃ¡rquicos:

- Estudia las entradas y salidas para averiguar quÃ© procesos genera.

- La estructura de control del programa debe ser jerÃ¡rquica y se debe derivar de la estructura de datos del programa.
- En el proceso de diseÃ±o: 1. Se definen las estructuras de los datos de entrada y salida, 2. se mezclan todas en una estructura jerÃ¡rquica de programa, 3. se ordena detalladamente la lÃ³gica procedimental para que se ajuste a esa estructura.
- El diseÃ±o lÃ³gico debe preceder y estar separado del fÃ−sico.

    - MetodologÃ−as Orientadas a datos NO jerÃ¡rquicos:

- Se considera a los datos como el "nÃºcleo" del SI.
- Con este enfoque, todos los sistemas construÃ−dos estÃ¡n integrados sobre un mismo modelo de datos.

- **MetodologÃ−as Mixtas:**

    - No se referencian en el libro.

SISTEMAS DE TIEMPO REAL

- Conceptos.
- CaracterÃ−sticas.
- Requisitos.

- **Conceptos:**

    - Son sistemas muy dependientes del tiempo.
    - Procesan informaciÃ³n mÃ¡s orientada al control que a los datos.
    - Son controlados por eventos externos.
    - Hay que distinguir entre "sistemas en tiempo real" y "sistemas en lÃ−nea". Los sistemas interactivos (en lÃ−nea) interactÃºan con *personas* mientras que los sistemas en tiempo real interactÃºan con *personas* y *dispositivos fÃ−sicos externos*.

- **CaracterÃ−sticas:**

    - Se lleva a cabo el proceso de *muchas actividades* simultÃ¡neamente ("concurrencia").
    - Se asignan *prioridades* a determinados procesos.
    - Se *interrumpe* una tarea antes de que concluya para comenzar otra de mayor prioridad.
    - Existe *comunicaciÃ³n* entre tareas (resultado tarea nueva tarea).
    - *Acceso simultÃ¡neo* a datos comunes (en memoria y en almacenamiento secundario).

- **Requisitos:**

    - Manejo de inetrrupciones.
    - ComunicaciÃ³n y sincronizaciÃ³n entre tareas.
    - Gestionar procesos concurrentes.
    - Dar respuesta oportuna y `a tiempo' ante eventos externos.
    - Datos continuos o discretos.

- **PRINCIPALES METODOLOGÃ AS DE DESARROLLO**

- MERISE.
- SSADM.

- MÃ TRICA.

METODOLOGÃ A MERISE

- Francesa
- Ciclo de vida mÃ¡s largo que los existentes anteriormente.
- IntroducciÃ³n de dos ciclos complementarios: **de abstracciÃ³n** y **de decisiÃ³n**.

METODOLOGÃ A SSADM

- Inglesa.
- Ã nfasis en los usurios: sus requisitos y su participaciÃ³n.
- DefiniciÃ³n del proceso de producciÃ³n: quÃ© hacer, cuÃ¡ndo y cÃ³mo.
- Tres puntos de vista: datos, eventos y procesos.
- MÃ¡xima flexibilidad en herramientas y tÃ©cnicas de implementaciÃ³n.
- **NO** cubre aspectos como la planificaciÃ³n estratÃ©gica ni la construcciÃ³n de cÃ³digo.

METODOLOGÃ A MÃ TRICA

- EspaÃ±ola.
- Objetivo: crear un marco metodolÃ³gico comÃºn para la palnificaciÃ³n y desarrollo de la AdministraciÃ³n PÃºblica EspaÃ±ola.
- Se enfoca directamente al **desarrollo**.

- El libro hace referencia a la versiÃ³n 2.0 de MÃ TRICA, por tanto es absolutamente improbable que Inma pregunte algo de lo que en Ã©ste se describe (exceptuando cualquier cuestiÃ³n relativa los tres puntos anteriores). AsÃ– que… FIN!

“POSIBLES” PREGUNTAS CAPÃ TULO III

1. El anÃ¡lisis estructurado se diferencia del clÃ¡sico en:

- ♦ Emplear un mÃ©todo de particionamiento efectivo.
- ♦ Construir un modelo lÃ³gico del sistema.
- ♦ Definir los procesos.
- ♦ Definir las lÃ–neas de diseÃ±o.

2. En el anÃ¡lisis estructurado:

- ♦ El texto se introduce en todos los detalle inmediatamente.
- ♦ Se va de lo abstracto al detalle, es grÃ¡fico y unidimensional.
- ♦ Se usa un mÃ©todo para particionar exclusivamente problemas complejos.
- ♦ Ninguna de las anteriores.

3. Una metodologÃ–a es un conjunto de componentes que especifican:

- ♦ CÃ³mo se debe dividir un proyecto en etapas y las tareas de cada etapa.
- ♦ QuÃ© estÃ¡ndares se van a utilizar en el desarrollo.
- ♦ El modelo de ciclo de vida a utilizar.

RESPUESTAS

1	2	3
a	d	a

CAPITULO V : GESTIÃ N DE PROYECTOS SOFTWARE

- DefiniciÃ³n de proyecto.
- PlanificaciÃ³n.

• PLANIFICACIÃ N - CONCEPTOS GENERALES

- Plan de Proyecto.
- GestiÃ³n de Compromisos.

- Conceptos.
- CaracterÃ–sticas del Plan de Proyecto.
- Elementos del proyecto.

CONCEPTOS GENERALES

- Consiste en el "estudio de etapas, actividades y tareas necesarias para alcanzar un objetivo que implica un trabajo no inmediato a un plazo relativamente largo"
- La GestiÃ³n de Proyectos aborda distintos aspectos: **planificaciÃ³n**, **control, direcciÃ³n** y **organizaciÃ³n**.
- El director de proyecto debe **planificar, controlar y dirigir** las actividades del proyecto.
- Las unidades organizativas se encargan de **organizar** las actividades del proyecto.
- La PlanificaciÃ³n de proyectos ha de cubrir: la realizaciÃ³n de un **plan de proyecto** (por parte del director de proyecto) y, la **gestiÃ³n de compromisos**.
- La funciÃ³n principal del director de proyecto es la realizaciÃ³n del plan de proyecto.

CARÃ"CTERÃ STICAS DEL PLAN DE PROYECTO

- Proporcionar un resumen del proyecto a altos directivos.
- Permitir que el director del proyecto y los clientes puedan supervisar el progreso del proyecto desde el inicio hasta el final.
- Debe ser un documento orientado al cliente.
- Debe ser un documento base que cuente con la aprobaciÃ³n del cliente y ademÃ¡s sea actualizable.

ELEMENTOS ESENCIALES DEL PROYECTO

- Resumen del proyecto. Comprendido por cualquier persona. Que indique los productos entregables.
- Lista de hitos alcanzables.
- Procedimientos y estÃ¡ndares a aplicar.
- EspecificaciÃ³n del proceso de revisiÃ³n para determinar `quiÃ©n', `cÃ³mo', `cÃºando' y `para quÃ©' se revisa el proyecto.
- Plan que defina la comunicaciÃ³n entre la organizaciÃ³n de desarrollo y el cliente.
- Diagrama de descomposiciÃ³n del trabajo (WBS).
- Lista del personal del proyecto y asignaciÃ³n en relaciÃ³n al WBS.
- Red de actividades que muestre la secuencia de activiades en el tiempo y las relaciones entre actividades.
- Presupuestos y calendarios para todas las actividades que tienen un responsable.

• PLANIFICACIÃ N - ACTIVIDADES PARA LA PLANIFICACIÃ N DE UN PROYECTO

- Plan de desarrollo.
- CaracterÃ–sticas de un Programa de Tiempos efectivo.
- Pasos para la realizaciÃ³n de un calendario.

PLAN DE DESARROLLO

- El objetivo principal del plan de desarrollo es la configuraciÃ³n del **calendario de proyecto** (o "programa de tiempos").

- El calendario de proyecto consiste en una representaciÃ³n grÃ¡fica de todas las actividades del proyecto, necesarias para el resulado final.
- El calendario permite al director de proyecto coordinar de forma efectiva al equipo de desarrollo.
- Un calendario debe ser `dinÃ¡mico' (que pueda variar).

CARÃ"CTERÃ STICAS DE UN PROGRAMA DE TIEMPOS

- Comprensible por las personas que van a utilizarlo.
- Suficientemente detallado para que pueda servir de base para medir y controlar el progreso del proyecto.
- Capaz de seÃ±alar actividades crÃ–ticas.
- Flexible (`fÃ¡cilmente modificable').
- Basados en estimaciones de tiempo fidedignas.
- Ajustable a recursos disponibles.
- Compatible con los planes de otros proyectos que compartan los mismos recursos.

PASOS PARA LA REALIZACIÃ N DE UN CALENDARIO

- **1. DefiniciÃ³n de los objetivos del proyecto:**

- Objetivo del proyecto: "enunciado que especifica los resultados que se desean conseguir".
- Los objetivos deben definirse al comienzo para identificar responsbilidades del equipo de desarrollo y se revisarÃ¡n durante el poyecto para seÃ±alar los cambios que se aparten del alcance inicial. Esto ayudarÃ¡ al director de proyecto a revisar los resultados finales respecto a los objetivos iniciales.

…Para que el objetivo quede *bien definido* tiene que ser:

- Asequible:
- Definitivo: especificar `quÃ© lograr' y `con quÃ© grado de detalle'.
- Cuantificable: especificar el criterio de finalizaciÃ³n.
- De duraciÃ³n especÃ–fica: especificar la duraciÃ³n de las actividades.

- **2. DescomposiciÃ³n de las actividades:**

- Para que el director de proyecto pueda:

    - ♦ Aumentar la supervisiÃ³n de trabajos.
    - ♦ Observar el seguimiento de actividades crÃ–ticas de gran repercusiÃ³n.

- **3. RelaciÃ³n entre las actividades:**

- Deben determinarse las secuencias y dependencias entre las actividades.

- **4. EstimaciÃ³n de tiempos y costes de las actividades:**

- **5. Reajuste del programa de tiempos a las restricciones del proyecto:**

- Objetivos principales:

    - ♦ Determinar la duraciÃ³n total del proyecto.
    - ♦ Identificar las activiades que contribuyen a la duraciÃ³n total del proyecto (“actividades crÃ–ticas”).
    - ♦ Calcular las holguras de las actividades NO crÃ–ticas.

- Excepto en proyectos muy sencillos, se hace necesario el uso de herramientas automatizadas que realicen el anÃ¡lisis de los tiempos de red y la distribuciÃ³n de los recursos pertinenentes.

- **6. AsignaciÃ³n de recursos / DefiniciÃ³n de la organizaciÃ³n del equipo:**

- Objetivos principales:

    - ♦ Ajustar el calendario respecto a los recursos disponibles.
    - ♦ Ajustar las fechas de comienzo de algunas actividades NO mcrÃ–ticas.
    - ♦ Suavizar las cargas de trabajo.
    - ♦ Ajustar costes al presupuesto.

- **7. RevisiÃ³n del calendario:**

- Para determinar si es o no realista.

- **PLANIFICACIÃ N - GESTIÃ N DE COMPROMISOS**

    - Es importante que los directivos tomen las decisiones y adopten los compromisos despues de que el grupo de software haya emitido sus opiniones sobre si los compromisos se consideran o no factibles.
    - Si los directivos se deciden a adoptar un compromiso poco factible, deben estar preparados para un posible aumento de los costes o un retraso en la fecha de entrega.

- **PLANIFICACIÃ N - TÃ CNICAS PARA LA REALIZACÃ N DE UN CALENDARIO**

    - Diagrmas de Hitos.
    - Diagramas de Gantt.
    - Programas de tiempos Full Wall.
    - Redes de precedencia.
        - ♦ Pert.
        - ♦ CPM.

DIAGRAMAS DE HITOS

- Es el mÃ©todo mÃ¡s simple para la realizaciÃ³n de un calendario.
- Consiste en un cuadro con dos columnas: en la primera se especifican las actividades y en la segunda las fechas de finalizaciÃ³n.
- Ventajas:

- ♦ Facilidad de uso.
- ♦ MÃ–nimo coste de preparaciÃ³n.

- Desventajas:
  - ♦ Incertidumbre sobre las fechas de comienzo de las actividades.
  - ♦ Imposibilidad de reflejar interrelaciones entre actividades.

Actividad

Fecha finalizaciÃ³n

DIAGRAMAS DE GANTT

- Se usa normalmente para proyectos pequeÃ±os (menos de 25 actividades).
- Es posiblemente el tipo de calendario mÃ¡s utilizado.
- No es posible representar dependencias entre actividades.
- Hace mÃ¡s sencillo representar el solapamiento entre actividades.
- Consiste en un diagrama de barras donde se hace una referencia cruzada entre las tareas (“filas”) y su tiempo de duraciÃ³n (“columnas”).

1

2

3

4

5

6

7

8

9

10

Tarea1

Tarea2

Tarea3

Tarea4

Tarea5

Tarea6

Tarea7

PROGRAMA DE TIEMPOS FULL WALL

- Parte de una reuniÃ³n donde intervienen todas personas involucradas en el proyecto.

- El director de proyecto desarrolla un diagrama de hitos y una lista de actividades donde se indica los responsables de cada una.
- Cada actividad se escribe en dos tarjetas (“inicio” y “final”).
- Cada persona clava la tarjeta en la pared en la semana de su elecciÃ³n.

- Alto grado de interacciÃ³n entre los participantes de la reuniÃ³n.
- No muestra claridad en los camminos crÃ–ticos.

REDES DE PRECENCIA: PERT Y CPM

- Conceptos generales.
- Casos en los que es conveniente su utilizaciÃ³n.
- Diferencias.
- Reglas.
- Holguras de los diagramas PERT.

- **Conceptos generales:**

- Red: “modelo grÃ¡fico que seÃ±ala las relaciones secuenciales entre los sucesos clave del proyecto”.

- PERT y CPM pueden mostrar el **camino crÃ–tico** (“secuencia mÃ¡s larga de las actividades conectadas a travÃ±es de la red, y que determina la duraciÃ³n total del proyecto”).
- Para disminuir el tiempo total deben reducirse los tiempos de las actividades dentro del camino crÃ–tico.

- **Conviene utilizar las Redes de Precedencia cuando un proyecto:**

- Todas las actividades estÃ¡n bien definidas.
- Las actividades se puedan comenzar, interrumpir y realizar de forma separada respecto a la secuencia dada.
- Las actividades se puedan relacionar unas con otras.
- Las actividades estÃ©n ordenadas (de modo que se pueda seguir una secuencia).
- Una vez comenzada una actividad, debe continuar sin interrupciÃ³n hasta su finalizaciÃ³n.

- **Principales diferencias entre PERT y CPM:**

- **CPM** se centra en las **actividades** y **PERT** en los **eventos** o sucesos. Esto da la ventaja a PERT, por considerar los eventos como los `hitos del proyecto', lo que favorece el control de la gestiÃ³n.
- **PERT** permite el **tratamiento de la probabilidad** para la estimaciÃ³n de tiempos. **CPM no**.

- **Reglas en el desarrollo de PERT y CPM:**

- Como mÃ–nimo, la red debe poseer 20 eventos
- Las redes realizadas manualmente deben contar con un mÃ¡ximo de 300 sucesos.
- Los proyectos que justifican un gran nÃºmero de actividades o eventos poseen las siguientes caracterÃ–sticas:

- Muy crÃ–ticos.
- Poseen un alto riesgo o incertidumbre.
- Involucran a muchas personas u organizaciones.
- TÃ©cnicamente complejos.
- Con actividad en diversas localidades geogrÃ¡ficas.

- **Holguras de los diagramas PERT:**

• **Holgura**	NÃºmero de unidades de tiempo que un *suceso* puede retrasar su finalizaciÃ³n SIN aumentar la duraciÃ³n total del proyecto.	**H**i = **TL**i - **TE**i
• **Holgura total**	NÃºmero de unidades de tiempo que la realizaciÃ³n de una *actividad* puede retrasar su finalizaciÃ³n, respecto al tiempo PERT previsto, sin aumantar la duraciÃ³n total del proyecto.	**HT**ij = **TL**j - **TE**i - **T**ij
• **Holgura libre**	Parte de la holgura total que puede consumirse sin afectar a las actividades siguientes.	**HL**ij = **TE**j - **TE**i - **T**ij
• **Holgura independiente**	Cantidad de holgura disponible si todas las actividades han comenzado en sus `tiempos last'.	**HI**ij = **TE**j - **TL**i - **T**ij

- **MÃ TODOS DE ESTIMACIÃ N DE COSTES**

- Todos los metodos de estimaciÃ³n de costes dependen de la **informaciÃ³n**.
- Los mÃ©todos estÃ¡n infuÃ–dos por el **tiempo** y el **sueldo del trabajador**.
-

DINERO

(coste del proyecto)

estimado por

SALARIOS DE LOS TRABAJADORES

- OpiniÃ³n de expertos.
- EstimaciÃ³n por analogÃ–a.
- DescomposiciÃ³n.
- Ecuaciones o modelos de estimaciÃ³n.
- Recomendaciones.

OPINIÃ N DE EXPERTOS

- CaracterÃ–sticas.
- TÃ©cnicas.

- **CaracterÃ–sticas:**
    - ♦ Muy subjetiva.
    - ♦ Permite contemplar simultÃ¡neamente distintos proyectos anteriores junto con el actual.

- **TÃ©cnicas para sintetizar la OpiniÃ³n de Expertos:**

Pasos:

- Los expertos ofrecen su punto de vista.
- Se realiza la media de las valoraciones.
- Se organiza una reuniÃ³n con el propÃ³sito de llegar a un consenso.

    - ♦ "**Delphi**": variante que consiste en calcular la media de las valoraciones de `expertos anÃ³nimos'. La media es devuelta a cada uno de ellos. A continuaciÃ³n, se vuelve a opinar sobre la media hasta lograr un consenso.

ESTIMACIÃ N POR ANALOGÃ A

- Consiste en tomar un proyecto similar para que sirva de base.
- Se basa en una `experiencia real'.
- Cosntituye un complemento a la opiniÃ³n de expertos.
- Es difÃ–cil conocer el grado de similitud con el sistema a comparar.
- El tamaÃ±o del tiempo y el coste del proyecto NO guardan una ecuaciÃ³n lineal (por lo cual no pueden calcularse mediante la estimacion por analogÃ–a).

DESCOMPOSICIÃ N

- Se divide el proyecto en etapas y se estima el coste de cada una de ellas.
- Sigue un enfoque Botton-Up (abajo-arriba).
- La estimaciÃ³n es personal del responsable de cada componente. Esto la hace mÃ¡s fiable.
- No contempla las nuevas actividades que se puedan precisar.

ECUACIONES O MODELOS DE ESTIMACIÃ N

- Consiste en lograr las estimaciones mediante fÃ³rmulas matemÃ¡ticas.
- **Modelos de coste**: estimaciones directas del esfuerzo o de la duraciÃ³n. Suelen ser X = CTE * Y.
- **Modelos de restricciones**: relacionan en el tiempo dos o mÃ¡s parÃ¡metros de coste.

ENFOQUE RECOMENDADO

- Realizar un juicio de expertos mediante Delphi en la negociaciÃ³n de contratos.
- Valorar la similitud con proyectos anteriores.
- Crear un modelo propio.
- Refinar a medida que se avanza en el desarrollo.

- **TEORÃ A RESTANTE**

- Siguiendo el libro, nos encontramos con otra posible clasificaciÃ³n de los modelos de estimaciÃ³n:
    - ♦ Modelos empÃ–ricos: basados en la opiniÃ³n de expertos y la estimaciÃ³n tanto Top-Down como Botton-Up, apoyada en datos histÃ³ricos.
    - ♦ Modelos estadÃ–sticos (lineales y no lienales): obtenidos mediante el anÃ¡lisis de regresiÃ³n estadÃ–stica.

- ♦ Basados en una teorÃ–a: sobre el esfuerzo de desarrollar software (y comprobaciones de datos) (P ej: SLIM).
- ♦ Modelos compuestos: combinan varias tÃ©cnicas (P ej: COCOMO).

- Sigue la explicaciÃ³n de cÃ³mo realizar los modelos SLIM y COCOMO.
- Hasta el final del capÃ–tulo, otros puntos a tener en cuenta son:
  - ♦ El enfoque recomendado para la estimaciÃ³n de costes (4 puntos).
  - ♦ Los objetivos que pretende conseguir el **seguimiento y la supervisiÃ³n del proyecto** (3 puntos).
  - ♦ Las Ã¡reas de riesgo quedebe tratar un director de proyecto en la **gestiÃ³n de riesgos** del software (7 puntos).

- …Inma apenas ha incidido en estos puntos, pero como vienen en el libro, aviso de que estÃ¡n ahÃ– por si acaso. Si nos sobra tiempo y tenemos ganas, nos los miramos tambiÃ©n.

"POSIBLES" PREGUNTAS CAPÃ TULO V

1. La GestiÃ³n de Proyectos Software:

- ♦ Estudio de etapas, actividades y tareas necesarias para alcanzar un objetivo que implica un trabajo no inmediato a un plazo relativamente largo, segÃºn distintos aspectos como PlanificaciÃ³n, Control, DirecciÃ³n y OrganizaciÃ³n.
- ♦ Conjunto de personas, entre ellas el jefe de proyecto y desarrolladores que trabajan para obtener un producto software.
- ♦ PlanificaciÃ³n del desarrollo del software en cuanto a funciones que debe realizar.

- La PlanificaciÃ³n de un Proyecto Software incluye dos aspectos importantes:
  - ♦ Plan de Proyecto y GestiÃ³n de Compromisos.
  - ♦ Plan de Proyecto y Calendario.
  - ♦ Calendario y DescomposiciÃ³n de Trabajos del Proyecto (WBS)

- Son tÃ©cnicas para realizar un calendario:
  - ♦ Redes de Precedencia.
  - ♦ Diagramas de Gantt y Diagramas de Hitos.
  - ♦ Todas son correctas.

- Las tÃ©cnicas de construcciÃ³n de calendario que **no** muestran las interrelaciones entre las distintas actividades:
  - Redes de Precedencia.
  - Diagramas de Gantt y Diagramas de Hitos.
  - Todas son correctas.
  - La GestiÃ³n de Compromisos:
    - ♦ Los directivos deben tomar decisiones una vez emitida la opiniÃ³n del grupo de software.
    - ♦ Forma parte de la EstimaciÃ³n de Costes.
    - ♦ Todas son correctas.
    - ♦ Los mÃ©todos de EstimaciÃ³n de Coste:
      - ◊ OpiniÃ³n de Expertos, EstimaciÃ³n por AnalogÃ–a y DescomposiciÃ³n.
      - ◊ Ecuaciones o Modelos de EstimaciÃ³n.
      - ◊ Todas son correctas.

◊ La primera versiÃ³n del Plan de proyecto incluye:

· QuÃ©, CÃ³mo, QuiÃ©n y CuÃ¡ndo se hace.
· Herramientas y TÃ©cnicas de GestiÃ³n y de Desarrollo.
· Todas son correctas.

RESPUESTAS

1	2	3	4	5	6	7
a	a	c	b	a	c	c

CAPITULO VI : ANÃ"LISIS DE NECESIDADES Y ESTUDIO DE VIABILIDAD

· **CÃ MO COMIENZA UN PROYECTO**

- Inicio a nivel de empresa.
- Inicio a nivel de proyecto.

INICIO A NIVEL DE EMPRESA

- Es el precedente necesario del inicio a nivel de proyecto.
- Elementos principales:
- DecisiÃ³n de emprender el proyecto.
- SelecciÃ³n del jefe de proyecto (responsable de establecer el entorno inicial de trabajo).

♦ **DecisiÃ³n de emprender el proyecto:**

- OrÃ–gnes de la idea de emprender un proyecto.
- Conceptos.
- AnÃ¡lisis de Viabilidad.
- OrÃ–genes de la idea de emprender un proyecto:

◊ En el caso de un **departamento de desarrollo** de la empresa:

- La constataciÃ³n de los requisitos del software estÃ¡ cambiando.
- PeticiÃ³n especÃ–fica del cliente o usuario.
- Propuesta desde dentro de la propia organizaciÃ³n de desarrollo.
- Necesidad detectada por el departamento de marketing.
- RecomendaciÃ³n especÃ–fica del presonal de mantenimiento.
- Detectada a partir de la informaciÃ³n de los usuarios.
- Primero se debe obtener una idea clara de lo que se pretende hacer, para despuÃ©s evaluar la viabilidad del proyecto.

◊ En el caso de una **empresa de servicios informÃ¡ticos**:

- Respuesta a una demanda del cliente.
- Conceptos generales:
- Es necesario definir y analizar los requisitos del sistema que se debe construir.
- Para lograr mejores resultados se debe obtener de los usuarios, informaciÃ³n sobre las necesidades que desean que el software satisfaga.
- Los resultados de estudios previos se indican en un documento llamado **"informe de necesidades"**.
- Consideraciones para analizar la viabilidad del proyecto:

- Considerar diferentes alternativas que se puedan concebir (comprar un software comercial ya creado, desarollar el producto internamente…).
- EvaluaciÃ³n de cada alternativa.
- EspecificaciÃ³n detallada de la alternativa desarrollada.
- Establecimiento de fechas y compormisos de trabajo por parte de las personas y los departamentos implicados (`definiciÃ³n de un plan alternativo para el proyecto').

♦ **SelecciÃ³n del director del proyecto:**

- Conceptos.
- CaracterÃ–sticas.
- Conceptos generales:
- Normalmente, la elecciÃ³n del director de proyecto es posterior la decisiÃ³n de emprender el proyecto.
- Seguramente sea el elemento mÃ¡s crÃ–tico de cara a conseguir el Ã©xito del proyecto.
- CarÃ¡cterÃ–sticas deseables en un director de proyecto:
- **Liderazgo**: habilidad para motivar a los componentes del equipo de proyecto.
- **ComprensiÃ³n tÃ©cnica**: conocimientos para tomar las decisiones correctas.
- **Competencia en la gestiÃ³n**: capacidad para controlar las activiades, costes y presupuestos del proyecto.
- **Presteza y decisiÃ³n**: para observar, evaluar y decidir.
- **Versatilidad y flexibilidad**: para encauzar acontecimientos imprevistos.
- **Integridad**: para reclutar al personal mÃ¡s adecuado y ganarse la confianza del cliente.
- **PrevisiÃ³n**: para anticiparse al problema y aportar soluciones.

INICIO A NIVEL DE PROYECTO

- **Establecer el entorno inicial de trabajo del proyecto** (por parte del director de proyecto).
- Definir el soporte necesario para el equipo de proyecto.
- Definir las tÃ©cnicas de comunicaciÃ³n entre miembros.
- Definir los requisitos que deben cumplir los posibles subcontratistas.
- Definir la manera de aboradar la interacciÃ³n con el cliente.

· **ESTUDIOS DE VIABILIDAD**

- Aspectos a abordar.
- AnÃ¡lisis Coste-Beneficio.
- Aspectos para anÃ¡lisis coste-beneficio eficaz.

ASPECTOS A ABORDAR EN EL ESTUDIO DE VIABILIDAD

- **EconÃ³mico**: si el beneficio compensa los costes.
- **TÃ©cnico**: si la funcionalidad, el rendimiento o las restricciones son razonables.
- **Legal**: si los requisitos atentan contra alguna ley o reglamento.
- **Operativo**: si se puede implantar de forma efectiva en la empresa.
- Se debe prestar especial atenciÃ³n al anÃ¡lisis econÃ³mico. Sus

conclusiones son determinantes para continuar o cancelar el proyecto.

ANÃ"LISIS COSTE-BENEFICIO

- Costes.
- Beneficios.
- Conceptos.
- **Costes a tener en cuenta:**

- Coste del personal informÃ¡tico implicado en el proyecto (desde el anÃ¡lisis hasta la instalaciÃ³n del sistema).
- Coste de consultorÃ–a.
- Coste de software adicional.
- Coste de hardware.
- Coste de infraestructura.
- Costes debidos al usuario.
- Costes contÃ–nuos: mantenimiento del sistema y alquileres.
- **Beneficios:**

- Nuevas funcionalidades.
- EliminaciÃ³n de errores.
- ReducciÃ³n de errores.
- Aumento de la velocidad.
- Aumento de la fiabilidad.
- **Conceptos:**

- Uno de los mayores problemas del anÃ¡lisis econÃ³mico es que ha de ser **realista**.
- En la mayorÃ–a de los casos la valoraciÃ³n de beneficios es necesariamente subjetiva.

ASPECTOS IMPORTANTES PARA QUE EL ANÃ"LISIS COSTE-BENEFICIO SEA EFICAZ

- La mayorÃ–a de las estimaciones de costes y beneficios deben consistir en **rangos de valores probables**.
- Es recomendable emplear **estimaciones pesimistas** o conservadoras.
- Se debe contar con **factores** que influyen en toda **prospectiva econÃ³mica**.
- Tratar de valorar y **prever** todos los **riesgo.**

· **TÃ CNICAS DE RECOGIDA DE INFORMACIÃ N**

- "Medio para mejorar la comunicaciÃ³n entre los usuarios o clientes y los desarrolladores de software".
- Pasos del proceso de AnÃ¡lisis.
- TÃ©cnicas de recogida de informaciÃ³n.
- Entrevistas.
- JAD.
- Prototipado.

PASOS DEL PROCESO DE ANÃ"LISIS

· **Identificar fuentes de informaciÃ³n** (usuarios) relevantes para el proyecto.

· **Realizar preguntas adecuadas** para comprender las necesidades.
· **Analizar la informaciÃ³n** recogida para detectar aspectos que quedan poco claros.
· **Confirmar con los usuarios** lo comprendido de los requisitos.
· **Sintetizar requisitos** en un documento de especificaciÃ³n apropiado

- El resultado del proceso deberÃ–a ser un documento que especifique lo mÃ¡s claramente posible, los requisitos que debe cumplir el software.

TÃ CNICAS PRINCIPALES DE RECOGIDA DE INFORMACIÃ N

Entrevistas	• Quiza la tÃ©cnica mÃ¡s empleada y la que require mayor preparaciÃ³n por parte del analista.
Desarrollo conjunto de aplicaciones (JAD)	• Se crean equipos de usuarios y analistas para determinar las caracterÃ–sticas que debe tener el software. • Cuenta con una mayor probabilidad de Ã©xito, ya que involucra al usuario en el proyecto de modo que lo aprecia como `algo propio'.
Prototipado	• ConstrucciÃ³n de un modelo o maqueta del sistema que permite al usuario evaluar mejor las necesidades.
ObservaciÃ³n	• Analizar `in situ' cÃ³mo funciona la unidad o departamento que se desea informatizar.
Cuestionarios	• Para recoger informaciÃ³n de un gran nÃºmero de personas en poco tiempo.
Estudio de documentaciÃ³n	• Para hacerse una idea sobre la normativa que rige la empresa.
Tormenta de ideas	• ReuniÃ³n de 4 a 10 personas. · Sugieren cualquier idea sin juzgar su validez. · Realizan un anÃ¡lisis detallado de cada propuesta.
ETHICS	• Para fomentar la perticipaciÃ³n de usuarios en varios proyectos. • Objetivo: satisfaciÃ³n de los empleados en el trabajo, mediante estudios integrales.

- Es una prÃ¡ctica habitual utilizar combinaciones de distintas tÃ©cnicas.

LAS ENTREVISTAS

- Conceptos.
- Cualidades del entrevistador.
- PreparaciÃ³n.
- RealizaciÃ³n.
- AnÃ¡lisis.
- **Conceptos:**

- Entrevista: “intento sistemÃ¡tico de recoger informaciÃ³n de otra persona”.
- La preparaciÃ³n es esencial para cumplir sus objetivos.
- Los elementos principales son: el entrevistador y el entrevistado.
- Es muy importante la forma en que se plantea la conversaciÃ³n.
- **Cualidades del entrevistador:**
- Imparcial.
- Ponderado.
- Buen oyente: capaz de escuchar mostrando interÃ©s (muy importante).
- Cierto grado de habilidad en el trato.
- Cordialidad y accesibilidad.
- Paciencia.
- **1. PreparaciÃ³n:**
- Fase en la que el entrevistador debe **documentarse e investigar la situaciÃ³n de la organizaciÃ³n**.
- Es **esencial para** que la **entrevista sea eficaz**.
- La entrevista debe **centrarse en aspectos** del trabajo **no accesibles por otros medios**.
- Otra actividad importante es la **identificaciÃ³n de personas que se debe entrevistar** (para minimizar el nÃºmero necesario de personas a entrevistar).
- Por Ãºltimo se debe **incluÃ–r**: la **preparaciÃ³n del objetivo y el contenido de la entrevista** y, la **planificaciÃ³n del lugar y la hora** donde se desarrollarÃ¡ la entrevista (el lugar ha de ser confortable y la fecha y hora deben adaptarse a la agenda del entrevistado).
- **2. RealizaciÃ³n:**
- La realizaciÃ³n se considera “el nÃºcleo central de la entrevista”.
- Se distinguen tres etapas: apertura, desarrollo y terminaciÃ³n.
- Apertura.
- Desarrollo.
- TerminaciÃ³n.
- **Apertura:**

- El entrevistador se presenta e informa al entrevistado sobre la razÃ³n de la entrevista.
- Los primeros minutos son fundamentales para crear un ambiente confortable.
- La primera impresiÃ³n en el entrevistado es muy importante.
- **Desarrollo:**

- No deberÃ–a exceder de 2 horas.

- TÃ©cnicas durante el desarrollo:
- **Preguntas abiertas** en los primeros momentos para proseguir con otras mÃ¡s directas y cerradas.
- Utilizar **palabras y frases aropiadas** (evitando tecnicismos o palabras o frases que puedan perturbar la comunicaciÃ³n).
- **Asentir** mientras se escucha.
- **Repetir respuestas dadas** (transmite la sensaciÃ³n de que el interlocutor atiende y entiende. Esre recurso debe utilizarse moderadamente).

- **Pausas**, para ejercer presiÃ³n y obligar al entrevistado a hablar.
- Lo ideal es que el entrevistador hable un 20% del tiempo y el entrevistado un 80%..

- **TerminaciÃ³n:**

- Se finaliza recapitulando la entrevista, agradeciendo el esfuerzo al entrevistado y citÃ¡ndolo para una nueva entrevista si fuese necesario.
- Es importante dejar abierta la posibilidad de volver a contactar.
- Hay que convencer al entrevistado de que se le ha entendido.
- **3. AnÃ¡lisis:**
- Leer las notas, reorganizar la informaciÃ³n, etc
- Es importante evaluar cÃ³mo ha ido la entrevista asÃ– como los aspectos mejorables.

DESARROLLO CONJUNTO DE APLICACIONES (JAD)

- Conceptos.
- Razones base.
- Proceso JAD.
- **Conceptos generales:**
- Su finalidad es promover la coperaciÃ³n y el trabajo en equipo entre los usuarios y los analistas.
- Consisten en un conjuto de reuniones (2-4 dÃ–as) en las que participan tanto usuarios cualificados como analistas de software.
- JAD no se utiliza demasiado debido a que require una mayor preparaciÃ³n que las entrevistas y el ambiente o los mÃ©todos de trabajo convencionales en las empresas no facilitan este tipo de actividades.
- **Razones que sirven de base a JAD:**
- Las entrevistas requieren mucho tiempo. JAD precisa pocos dÃ–as.
- Es mÃ¡s difÃ–cil apreciar los posibles errores en la especificaciÃ³n de requisitos cuando sÃ³lo un analista revisa el documento. En el JAD todo el grupo puede actuar como revisor y detectar defectos.
- JAD propugna una participÃ ciÃ³n mÃ¡s profunda de los usuarios en el proyecto.
- **Proceso tÃ–pico JAD:**

1. AdaptaciÃ³n o preparaciÃ³n

Tareas de preparaciÃ³n de sesiÃ³n por parte del jefe del JAD:

- SelecciÃ³n de participantes (conjunto lo mÃ¡s representativo posible).
- Recabar cierta informaciÃ³n sobre el sistema a construÃ–r.
- Organizar la reuniÃ³n.

2. SesiÃ³n JAD

- Reuniones donde se parte de un documento de trabajo que hay que analizar para completar el conjunto de requisitos del sistema.
- Al final de la sesiÃ³n se habrÃ¡ concluÃ–do el documento de especificaciÃ³n, aprobado por los presentes

3. DocumentaciÃ³n

- Completar el documento final de sesiÃ³n.

EL PROTOTIPADO

- Conceptos.
- Casos en los que resulta Ãºtil.
- Razones.
- Herramientas.
- **Conceptos generales:**
- Consiste en la elaboraciÃ³n del modelo o maqueta del sistema para evaluar mejor los requisitos que se desea que cumpla el sistema.
- Una cualidad esencial del prototipo es su posibilidad de ser construÃ–do mÃ¡s rÃ¡pidamente que la aplicaciÃ³n.
- **Casos en los que resulta mÃ¡s Ãºtil:**
- Cuando el Ã¡rea de aplicaciÃ³n NO estÃ¡ bien definida.
- El coste de rechazo de la aplicaciÃ³n por parte de los usuarios es muy alto.
- Es necesario evaluar previamente el impacto del sistema en los usuarios y la organizaciÃ³n.
- **Razones principales para utilizar el prototipado:**
- (Ordenadas por frecuencia de uso).

1. Prototipado de la interfaz de usuario:	• Para asegurarse de que estÃ¡ bien diseÃ±ada y satisface las necesidades de los usuarios. • No es caro. • Muy frecuente. • Usualmente consiste en simples modelos de pantallas.
2. Modelos de rendimiento:	• Para evaluar el posible rendimiento del diseÃ±o.
3. Prototipado funcional:	• Prototipo: “primera versiÃ³n del sistema, con funcionalidad limitada”. • Tras comprobar si las funciones implentadas son apropiadas, se corrigen, refinan y aÃ±aden nuevas hasta obtener el sistema final.

- **Herramientas para la realizaciÃ³n del prototipado:**
- Nivel inferior herramientas comunes y facilmente disponibles (P ej: programas de dibujo o presentaciÃ³n, hojas de cÃ¡lculo o generadores de informes. Con Ã©stas se consigue que el usuario pueda ver la entrada/salida que se tendrÃ¡ cuando la aplicaciÃ³n estÃ© acabada.
- Nivel un poco mÃ¡s evolucionado algunos gestores de bases de datos.
- Otra opciÃ³n posibilidades de prototipados que proporcionan las herramientas CASE o determinados generadores de aplicaciones.

“POSIBLES” PREGUNTAS CAPÃ TULO VI

1. Un Estudio de Viabilidad debe abordar aspectos como:

- ♦ EconÃ³mico y TÃ©cnico.
- ♦ Legal y Operativo.
- ♦ Todas son correctas.

2. El encarrgado de establecer el entorno inicial de trabajo de un proyecto software:

- ♦ Jefe de Proyecto.
- ♦ Desarrolladores y usuarios.
- ♦ Directivos.

3. El Informe de Necesidades incluye:

- ♦ Resultados de Estudio Previo.
- ♦ La Viabilidad del Proyecto.
- ♦ Todas son correctas.

RESPUESTAS

1	**2**	**3**
c	a	a

CAPITULO VII : ANÃ"LISIS DE SISTEMAS

· **INTRODUCCIÃ N AL ANÃ"LISIS DE REQUISITOS**

- Definiciones.
- ClasificaciÃ³n de actividades.

DEFINICIONES

- **AnÃ¡lisis (aplicado a sistemas):** "descomponer el sistema en sus componentes para estudiar cada uno de ellos".
- **AnÃ¡lisis (aplicado a una fase del ciclo de vida):** "producir un documento de especificaciÃ³n de requisitos que describa lo que el futuro sistema debe hacer (pero NO cÃ³mo debe hacerlo)".
- **AnÃ¡lisis de Requisitos:** "proceso del estudio de las necesidades de los usuarios para llegar a una definiciÃ³n de los requisitos del sistema de hardware o de software". TambiÃ©n: "Proceso y estudio de dichos requisitos".
- **Requisito:** "Condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificaciÃ³n".
- Para obtener los requisitos se requiere del trabajo conjunto de: **suministradores**, **desarrolladores**, **clientes** y **usuarios**.

CLASIFICACIÃ N DE LAS ACTIVIDADES DEL ANÃ"LISIS DE REQUISITOS

- SegÃºn IEEE 1074.
- ClasificaciÃ³n alternativa.
- **SegÃºn el estÃ¡ndar IEEE 1074**

- Definir los requisitos del software.
- Definir los requisitos de las interfaces.
- Integrar los requisitos (en un documento de especificaciÃ³n) y asignarle prioridades.
- **Otra clasificaciÃ³n (by RAGHAVAN)**

Actividades	TÃ©cnicas que emplean
· ExtracciÃ³n o determinaciÃ³n de requisitos:	• TÃ©cnicas de recogida de informaciÃ³n (capÃ–tulo 6) (Ej: observaciÃ³n, cuestionarios, entrevistas…)
· AnÃ¡lisis de requisitos:	• TÃ©cnicas grÃ¡ficas. • Modelos competos (Ej: AnÃ¡lisis Estructurado)
· EspecificaciÃ³n de requisitos:	• TÃ©cnicas grÃ¡ficas. • Modelos competos (Ej: AnÃ¡lisis Estructurado)
· ValidaciÃ³n de requisitos:	• Listas de comprobaciÃ³n. • TÃ©cnicas de revisiÃ³n.

♦ Estas actividades NO tienen por quÃ© realizarse en secuencia.

· **ESPECIFICACIÃ N DE REQUISITOS DEL SOFTWARE**

- Definiciones.
- Definiciones ERS.
- CaracterÃ–sticas principales.
- CaracterÃ–sticas generales.
- EvoluciÃ³n.
- EspecificaciÃ³n de requisitos de interfaces.

DEFINICIONES

- **EspecificaciÃ³n:** “documento que define el diseÃ±o, el comportamiento u otras caracterÃ–sticas de un sistema o componente de un sistema”.
- **Software:** “conjunto de programas, procedimientos y documentaciÃ³n asociada a la operaciÃ³n de un sistema informÃ¡tico”.

DEFINICIONES DE ERS

- **Def 01:** “documentaciÃ³n de los requisitos esenciales del software y de sus interfaces externas”.
- **Def 02:** “documento en el que se integran los requisitos una vez superada su verificaciÃ³n y validaciÃ³n”.
- **Def 03:** “descripciÃ³n de `lo que hay que desarrollar', lo cual implica una descripciÃ³n correcta de todos los requisitos del software (evitando requisitos innecesarios) y, NO describir ningÃºn detalle del diseÃ±o de software (excepto las restricciones del diseÃ±o que influyen en los requisitos).
- Una ERS debe especificar SOLAMENTE `lo que desea el usuario'.

CARACTERÃ STICAS FUNDAMENTALES DE UNA ERS

- **InformaciÃ³n veraz**: coherente con las necesidades reales de los usuarios.
- **Comunicada de forma eficaz**: de modo que se pueda comprender perfectamente.

CARACTERÃ STICAS GENERALES PARA UNA BUENA ERS

- No ambigua.
- Completa.
- FÃ¡cil de verificar.
- Consistente.
- FÃ¡cil de modificar.
- Facilidad para identificar el origen y las consecuencias de cada requisito.
- Facilidad de utilizaciÃ³n en la fase de explotaciÃ³n y mantenimiento.
- **No ambigua:**

- Cada requisito ha de tener **una Ãºnica** interpretaciÃ³n.
- **Completa:**

- Incluir todos los requisitos significativos del software.
- Definir la respuesta del software a todas las posibles clases de datos de entrada y en todas las situaciones posibles.
- Estar conforme con cualquier estÃ¡ndar de especificaciÃ³n a cumplir.
- Las figuras, tablas y diagramas han de estar etiquetadas y referenciadas en el texto.
- **By Bruce**: A continuaciÃ³n nos encontramos en el libro con la explicaciÃ³n de que existe un simpÃ¡tico recurso empleado en las ERS. Consistente en utilizar unas siglas: TBD (To Be Determined) que viene a significar “vamos, esto… ya lo haremos, tÃº tranquilo”, y que se aÃ±aden en las especificaciones cuando llegamos a un punto que aÃºn no tenemos claro. LÃ³gicamente nos cuenta que es un recurso `no demasiado elegante' y que cuando nos veamos obligados a usarlo, incluyamos tambiÃ©n la forma en que pensamos solucionar la situaciÃ³n en un futuro para asÃ– deshacernos del TBD. … No creo que vaya a caer en el examen, es una estupidez, asÃ– que… menciÃ³n hecha y… seguimos.
- **FÃ¡cil de verificar:**

- Cualquier requisito referenciado se pueda verificar fÃ¡cilmente.
- **Consistente:**

- NingÃºn conjunto de requisitos han de ser contradictorios o entrar en conflicto.
- Posibles conflictos:
- Empleo de distintos tÃ©rminos para el mismo objeto.
- CaracterÃ–sticas de objetos reales entran en conflicto.
- Conflicto lÃ³gico o temporal entre dos acciones determinadas
- **FÃ¡cil de modificar:**

- La estructura y el estilo deben permitir que cualquier cambio necesario en los requisitos se pueda realizar fÃ¡cil, completa y consistentemente.
- La ERS debe:
- Tener una organizaciÃ³n coherente y manejable (con tablas de contenidos, Ã–ndices y refernecias cruzadas).
- No ser redundante (el mismo requisito NO debe aparecer en mÃ¡s de un lugar de la ERS).

- La “redundancia” NO equivale a “error”, pero puede conducir a errores.

- **Facilidad para identificar el origen y las consecuencias de cada requisito (facilidad de traza):**

- Establecer el origen claro para cada uno de los requisitos.
- Posibilitar la referencia de los requisitos en desarrollos futuros o incrementos de la documentaciÃ³n.
- Referencias recomendas:
- Referencias hacia atrÃ¡s: a documentos previos.
- Referencias hacia delante: a documentos originados a partir del ERS.
- **FÃ¡cilidad de utilizaciÃ³n en la fase de explotaciÃ³n y mantenimiento:**

- Causas.
- Objetivos.
- Causas:
- El personal que NO ha estado relacionado con el desarrollo se encarga del mantenimiento.
- Gran parte de los conocimientos y de la informaciÃ³n se dan por supuestos, pero suelen estar ausentes.
- Objetivos:
- Ser fÃ¡cilmente modificable.
- Prever el registro de las caracterÃ–sticas especiales de cada componente (su criticidad, su relaciÃ³n con las necesidades temporales y su origen).

EVOLUCIÃ N DE LA ERS

- Conceptos.
- Causas.
- **Conceptos:**

- La ERS, normalmente, necesitarÃ¡ ser cambiada q medida que progresa el producto software.
- Muy posiblemente se realizarÃ¡n cambios adicionales como consecuencia de haber encontrado deficiencias.
- **Consideraciones a tener en cuenta en este proceso:**

- El requisito debe ser especificado de la forma mÃ¡s completa posible (aÃºn en el caso de que se prevean revisiones en el proceso de desarrollo).
- Debe iniciarse un proceso formal de cambio (para identificar, controlar, seguir e informar de cambios tan pronto como sean identificados).

ESPECIFICACIÃ N DE REQUISITOS DE LAS INTERFACES

- Interfaces con el exterior â “Entradas y salidas (E/S) del sistema”.
- La definiciÃ³n de las interfaces E/S tiene como objetivo la estabilizaciÃ³n del modo en que el sistema va a interactuar con el

exterior del sistema.

- Debe contarse ademÃ¡s de con lo que exprese el usuario, con criterios que le ayuden a decidir quÃ© es lo mejor para su trabajo.
- **VISIÃ N GENERAL DE LAS TÃ CNICAS DE ESPECIFICACIÃ N**
  - ♦ Intro.
  - ♦ SegÃºn forma de representaciÃ³n.
  - ♦ SegÃºn enfoque de modelizaciÃ³n.

INTRO

- ♦ No hay un criterio definitivo (trivial) por el que se puedan organizar las tÃ©cnicas.

CLASIFICACIÃ N SEGÃ N LA FORMA DE REPRESENTACIÃ N

- ♦ GrÃ¡ficas.
- ♦ Textuales.
- ♦ Marcos o plantillas.
- ♦ Matriciales.
- ♦ **GrÃ¡ficas:**
  - ◊ Utilizan una serie de iconos en el que cada uno representa un componente de un aspecto en particular del modelo.
  - ◊ Preferidas cuando es importante resaltar la conexiÃ³n entre los distintos componentes.
  - ◊ Se suelen utilizar en combinaciÃ³n con las demÃ¡s tÃ©cnicas.
- ♦ **Textuales:**
  - ◊ Emplea una gramÃ¡tica definida (mÃ¡s o menos formal).
  - ◊ Para especificar con mÃ¡s detalle los componentes definidos en los grÃ¡ficos.
  - ◊ Ej: pseudocÃ³digoempleado en las especificaciones de procesos.

♦ **Marcos o plantillas:** ◊ Se representan mediante un formulario en el que se incluyen todas las caracterÃ–sticas del componente por medio de campos en el que cada entrada puede tener una gramÃ¡tica particular. ◊ Especifican la informaciÃ³n relativa a un componente de un modelo que ha sido declarado en un diagrama o en otro marco.	**ENTIDAD**: Estudiante. **DESCRIPCIÃ N**: Un *estudiante* es cualquier persona que estÃ¡ matriculada en una de las asignaturas ofertadas. **ATRIBUTOS**: NÂº DNI, apellidos, nombre, direcciÃ³n provincia, telÃ©fono. **IDENTIFICADORES**: NÂº DNI. **RESTRICCIONES**:

	VOL. MAX. OCURRENCIAS: 2000. Ejemplo de plantilla de una entidad

- **Matriciales:**
  - ◊ No se consideran tÃ©cnicas de definiciÃ³n, sino de comprobaciÃ³n entre modelos.
  - ◊ Permiten estudiar referencias cruzadas entre los componentes de los modelos.

CLASIFICACIÃ N SEGÃ N EL ENFOQUE DE MODELIZACIÃ N

- **TÃ©cnicas para analizar un sistema. ClasificaciÃ³n 1**.
  - ◊ Atendiendo a las **funciones**: DFD.
  - ◊ Atendiendo a la **informaciÃ³n**: MER (E/R).
  - ◊ Atendiendo al **tiempo**: Lista de eventos.
- **TÃ©cnicas para analizar un sistema. ClasificaciÃ³n 2: segÃºn perspectivas**.
  - ◊ Plano **InformaciÃ³n-FunciÃ³n**: DFD.
  - ◊ Plano **FunciÃ³n-Tiempo**: Redes de Petri.
  - ◊ Plano **InformaciÃ³n-Tiempo**: Diagrama de Historia de Vida de la Entidad.

• MODELIZACIÃ N DE FUNCIONES

- **By Bruce 2:** Este apartado se corresponde en su mayorÃ–a a las instrucciones para realizar la "parte prÃ¡ctica". Por lo tanto he obviado `algunos ` puntos, lo cual ha hecho que la estructura del apartado no sea del todo lustrosa.

De todos modos, y como siempre: comprobar con el libro e incluir los apuntes extras que se consideren necesarios.

- Diagramas de Flujo de Datos.
- Diccionario de Datos.
- EspecificaciÃ³n de Procesos.
- Diagramas de DescomposiciÃ³n Funcional.
- Comprobaciones sobre una especificaciÃ³n estructurada.

- **Diagramas de Flujo de Datos (DFD):**
  - ◊ Componentes:
  - ◊ **Procesos**: componentes funcionales del sistema.
  - ◊ **Almacenes**: representan los datos almacenados o en reposo.
  - ◊ **Entidades externas**: representan las fuentes y destinos de la informaciÃ³n del sistema.
  - ◊ **Flujos de datos**: representan los datos que fluyen entre las funciones. Pueden ser "discretos" (representan datos en movimiento en un momento determinado), o "continuos" (flujos de datos persistentes en el tiempo).

- **Diccionario de Datos (DD):**

  - **DefiniciÃ³n:** “lista organizada de los datos utilizados por el sistema (que grÃ¡ficamente se encuentran representados por los flujos de datos y almacenes presentes sobre el conjunto del DFD).

- **EspecificaciÃ³n de Procesos:**

  - Objetivo.
  - MÃ©todos.
  - Objetivo:

- “Especificar `lo que hace el proceso'. CÃ³mo transforma las entradas en salidas”.
  - MÃ©todos para la EspecificaciÃ³n de Procesos:
  - Lenguaje Estructurado.
  - Ã”rboles de DecisiÃ³n.
  - Tablas de DecisiÃ³n.
  - Diagramas de AcciÃ³n.
- **Diagramas de DescomposiciÃ³n Funcional:**
  - Objetivo:
- “Representar la jerarquÃ–a de los procesos del sistema en diferentes niveles de abstracciÃ³n”.

- **Comprobaciones a efectuar sobre una especificaciÃ³n estructurada:**

  - Aspectos a tener en cuenta.
  - Apunte chorra final.
  - Aspectos a tener en cuenta:
  - **CompleciÃ³n**: si los modelos son completos.
  - **Integridad**: si NO existen contradicciones ni inconsistencias entre los distintos modelos.
  - **Exactitud**: si los modelos cumplen los requisitos del usuario.
  - **Calidad**: estilo, legibilidad y facilidad de mantenimiento de los modelos producidos.

  - Leer tabla 7.9 (PÃ¡g. 212 del libro) y si en el examen reconocemos alguna de las frases, marcamos la opciÃ³n “Es una pregunta empleada en la revisiÃ³n de una especificaciÃ³n estructurada”.

- Ejercicio prÃ¡ctico que nos sirve de ejemplo:

• La cuestiÃ³n “Â¿Hay almacenes locales?”:
  - OpciÃ³n que NO es.
  - Se utiliza para la revisiÃ³n de una especificaciÃ³n estructurada.
  - Esta tampoco es.
- Muy poco probable que aparezca en el examen una cuestiÃ³n sobre la susodicha tabla, pero… por si acaso… la leemos un

par de veces.

“POSIBLES” PREGUNTAS CAPÃ TULO VII

1. La EspecificaciÃ³n de Requisitos Software:

- ◊ DescomposiciÃ³n en componentes y estudio de cÃ³mo se realiza cada componente.
- ◊ Documento resultado del AnÃ¡lisis de un SI, describiendo quÃ© debe hacer dicho SI.
- ◊ Todas son correctas.

2. Requisito:

- ◊ CondiciÃ³n que necesita un usuario para resolver un problema o alcanzar un objetivo.
- ◊ Proceso de estudio de las necesidades de los desarrolladores.
- ◊ Todas son correctas.

3. Son tÃ©cnicas para el AnÃ¡lisis de Requisitos:

- ◊ GrÃ¡ficas como el Diagrama de Flujo de Datos.
- ◊ Entrevistas y ObservaciÃ³n para recogida de requisitos.
- ◊ Todas son correctas.

4. Las dos caracterÃ–sticas fundamentales de una EspecificaciÃ³n de Requisitos:

- ◊ InformaciÃ³nVeraz y Comunicada de forma Eficaz.
- ◊ FÃ¡cil de Modificar y FÃ¡cil de Verificar.
- ◊ Todas son correctas.

5. Las comprobaciones a realizar sobre una EspecificaciÃ³n estructurada:

- ◊ Si los modelos son Completos y Exactos con respecto a los requisitos impuestos.
- ◊ Si no existen Contradicciones ni Inconsistencia entre modelos y si son de Calidad en cuanto a estilo, legibilidad y facilidad de mantenimiento.
- ◊ Todas son correctas.

RESPUESTAS

1	**2**	**3**	**4**	**5**
B	a	b	a	c

CAPITULO VII - VIII :

- ANÃ”LISIS DE SISTEMAS -

- DISEÃ O ESTRUCTURADO DE SISTEMAS -

♦ Intro. ♦ DED.	By Bruce. AnÃ¡lisis de Sistemas.

♦ TÃ©cnicas de EspecificaciÃ³n de Control. ♦ Comprobaciones entre modelos. ♦ DistinciÃ³n chorra. ♦ Intro DiseÃ±o Estructurado. ♦ Atributos de Calidad. ♦ NormalizaciÃ³n. ♦ MetodologÃ–as de DiseÃ±o Detallado de Programas.	AnÃ¡lisis de Sistemas. AnÃ¡lisis de Sistemas. AnÃ¡lisis/DiseÃ±o. DiseÃ±o Estructurado de Sistemas. DiseÃ±o Estructurado de Sistemas. DiseÃ±o Estructurado de Sistemas. DiseÃ±o Estructurado de Sistemas.

INTRO BY BRUCE

A ver…

SÃ–, he mezclado apartados del Tema 7 con otros del Tema 8. Â¿Por quÃ©? … Las razones son varias, siendo la principal de ellas: "porque soy yo quien hace los resÃºmenes".

Realmente hay unas cuantas de razones mÃ¡s, pero no me voy a dedicar ahora a justificarme.

"CapÃ–tulo 78 by Bruce" y no hay mÃ¡s que discutir .

En otro orden de cosas. Una de las preguntas incluidas al final no puede responderse con este resumen. Esto es porque la pregunta es relativa al modelo relacional, que sÃ– aparece en el tema del libro, pero al que yo he decidido tratar aparte.

- **DIAGRAMA DE ESTRUCTURA DE DATOS (DED)**

♦ Es un modelo de datos mÃ¡s sencillo y de menor nivel de abstracciÃ³n que el E/R. ♦ Todas sus interrelaciones son 1:N. ♦ Mas que a nivel conceptual, este modelo se emplea para representar el modelo lÃ³gico de datos.	

- **TÃ CNICAS DE ESPECIFICACIÃ N DE CONTROL**
  - ♦ Lista de Eventos.
  - ♦ Diagramas de transiciÃ³n de Estados.
  - ♦ Redes de Petri.

LISTA DE EVENTOS

- Conceptos generales.
- Definiciones.
- Tipos de eventos.
- **Conceptos generales:**

- La lista de eventos consiste en un “listado de los eventos que se producen en el sistema”.
- Los eventos se obtienen de los DFD. A efectos prÃ¡cticos: un evento por cada funciÃ³n.
- **Definiciones:**

- Evento: “suceso que modifica el estado de los datos”.
- Disparador: lo que da lugar a que comience un evento. Suelen ser flujos de datos que entran en el sistema.
- **Tipos de eventos:**

- Generados externamente: si el disipador es un flujo que entra en el sistema.
- Reconocidos internamente: provocado por el estado interno de un dato. (Ej: cuando un dato toma un determinado valor).
- Basados en el tiempo: cada cierto tiempo se lanza el evento.

DIAGRAMA DE TRANSICIÃ N DE ESTADOS (DTE)

- Conceptos generales.
- Componentes.
- Aplicaciones.
- **Conceptos generales:**

- **Def:** “grÃ¡fico que recoge los distintos estados que puede tomar el SI”.
- Enfocado al “comportamiento del sistema en funciÃ³n del tiempo”.
- Se utiliza comunmente para comprobar el funcionamiento de *sistemas en tiempo real*.
- **Componentes:**

- Estados: representan los modos externos de comportamiento.
- Transiciones: obligan al paso de un estado a otro (o a sÃ– mismo) si se cumple una condiciÃ³n.
- **Aplicaciones:**

- Especificar transformaciones de datos.
- Especificar transformaciones de control.

REDES DE PETRI

- Conceptos generales.
- Componentes.
- Similitud con el DTE.
- **Conceptos generales:**

- Especialmente indicada para la descripciÃ³n de sistemas de

control **asÃ–ncronos** y **concurrentes**.

- **Componentes:**

- Conjunto finito de lugares.
- Conjunto diniro de transiciones.
- Conjunto finito de conexiones o arcos.
- **Similitud con el Diagrama de TransiciÃ³n de Estados:**

- Los DTE pueden considerarse como un determinado tipo de redes de Petri.
- Ambos modelos deben contar con las siguientes propiedades: **limitaciÃ³n** y **vivacidad**.
- **COMPROBACIONES ENTRE LOS DISTINTOS MODELOS DE ANALISIS**
  - Principales modelos.
  - Comprobaciones.
  - TÃ©cnicas.
  - Matriciales.
  - HVE.

PRINCIPALES MODELOS SOBRE LOS QUE EFECTUAR COMPROBACIONES

- **DimensiÃ³n de funciÃ³n**: DFD, DD y Especificaciones de Procesos (modelos de la EspecificaciÃ³n Estructurada).
- **DimensiÃ³n de informaciÃ³n:** E/R y Diagrama de Estructura de Datos.
- **DimensiÃ³n de tiempo:** Lista de Eventos.

COMPROBACIONES A REALIZAR ENTRE LOS MODELOS

- **Plano informaciÃ³n-funciÃ³n :**
  - Comprobar que todos los elementos (o datos elementales) definidos en los diagramas E/R estÃ¡n definidos en el DD.
  - Realizar la misma comprobaciÃ³n con los diagramas de estructuras de datos.
  - Comprobar que cada entidad e interrelaciÃ³n del E/R es consultada y actualizada al menos una vez por alguna funciÃ³n primitiva del DFD.
- **Plano informaciÃ³n-tiempo :**
- **Plano tiempo-funciÃ³n :**

- **By Bruce**: las comprobaciones entre modelos del plano informaciÃ³n-tiempo y tiempo-funciÃ³n involucran ciertas comprobaciones respecto a los diagramas HVE. Inma en su dÃ–a dijo: “Los HVE… con que sepÃ¡is que existen me vale.” (Sin embargo sÃ– han caÃ–do un par de preguntas sobre los HVE) Por lo tanto… bastante dudosa la

apariciÃ³n en el examen de alguna pregunta relativa a estos puntos. En todo caso, si no te fÃ–as demasiado, cogemos el libro y lo miramos tambiÃ©n (son cuatro chorradas contadas).

TÃ CNICAS PARA EFECTUAR COMPROBACIONES ENTRE MODELOS

- ◊ Matriciales.
- ◊ HVE.
- ◊ **TÃ©cnicas Matriciales:**
- ◊ Objetivo: se utilizan principalmente para ayudar a verificar la consistencia entre los compontes de los distintos modelos de un sistema.
- ◊ Matriz Entidad/FunciÃ³n: permite ver la relaciÃ³n existente entre las funciones del sistema y la informaciÃ³n necesaria para llevar a cabo cada una de las funciones.

Funciones	**Entidades**
Gestionar Presupuesto Cliente	
Gestionar Cliente	
…	
CLIENTE	**L**
	I, M, B
…	
PRESUPUESTO	
	I, M, B
…	
…	
…	

◊ Matriz Entidad/Entidad: permite ver las relaciones que tiene una entidad con otras del modelo E/R.

Entidad

Entidad

CLIENTE

PRESUPUESTO

CLIENTE

Realiza

PRESUPUESTO

◊ Matriz Evento/Entidad: muestra las alteraciones de las entidades, causadas por los eventos del sistema.

Entidades

Eventos

CLIENTE

PRESUPUESTO

Datos del Cliente

I, M, B

Datos del presupuesto

I

I, M, B

◊ **Modelado Evento/Entidad - HVE: Historia de Vida de la Entidad:**
  · Conceptos.
  · HVE en el anÃ¡lisis.
  · HVE en el diseÃ±o.
  · Conceptos:

◊ "Esta tÃ©cnica permite ver cÃ³mo las entidades han evolucionado en el tiempo".

◊ La HVE muestra el ciclo de proceso de una entidad desde su creaciÃ³n hasta su desapariciÃ³n.

◊ HVE utiliza la notaciÃ³n de diagrama de estructura.
  · HVE en el AnÃ¡lisis:
    • Proporciona una validaciÃ³n para los DFD y el modelo de datos.
  · HVE en el DiseÃ±o:
    • AquÃ– se aÃ±aden los indicadores

de estado a las HVE para permitir la validaciÃ³n semÃ¡ntica.

♦ **DISTINCIÃ N ENTRE ANÃ"LISIS Y DISEÃ O**

ANÃ"LISIS	DISEÃ O
Define **QUÃ** hacer. Son tÃ©cnicas empleadas en la fase de anÃ¡lisis: ◊ E/R (…relativa a datos). ◊ DFD (…relativa a funciones).	Define **CÃ MO** hacerlo. Se siguen dos enfoques: ◊ De Datos. ◊ Funcional.

8.1 DISEÃ O ESTRUCTURADO DE SISTEMAS - CONCEPTOS

◊ **Objetivo**: desarrollar la estructura de programa, asÃ– como las relaciones entre los elementos que componene dicha estructura ("mÃ³dulos").
◊ El diseÃ±o estructurado nos permite, partiendo del DFD, obtener una descripciÃ³n de la estructura del programa. Esta descripciÃ³n se representa mediante un **diagrama de estructuras**.

8.2 ATRIBUTOS DE CALIDAD DE UN DISEÃ O

◊ Acoplamiento.
◊ CohesiÃ³n.

ACOPLAMIENTO

◊ **DefiniciÃ³n:** "grado de interdependencia entre mÃ³dulos".
◊ Lo **Ã³ptimo** serÃ–a que el **acoplamiento** fuese **mÃ–nimo** (hacer los mÃ³dulos tan independientes los unos de los otros como sea posible).
◊ Para que el acoplamiento sea mÃ–nimo ningÃºn mÃ³dulo tiene que preocuparse de los detalles de la construcciÃ³n interna del resto de los mÃ³dulos.
◊ **Acoplamientos MÃ"S deseados:**

◊ 1.Acoplamiento normal: SÃ LO se produce una llamada entre los mÃ³dulos (no se pasan parÃ¡metros).
◊ 2.Acoplamiento de datos.
◊ **Acoplamientos MENOS deseados:**

◊ Uff 1.Acoplamiento por contenido: un mÃ³dulo hace referencia al interior de otro.
◊ Uff 2. Acoplamiento global: los mÃ³dulos comparten una estructura global de datos.

COHESIÃ N

◊ **DefiniciÃ³n:** “relaciÃ³n que existe entre los elementos de un mismo mÃ³dulo”.
◊ Lo **Ã³ptimo** es que la **cohesiÃ³n** sea **mÃ¡xima**.
◊ Deben agruparse en mÃ³dulos los elementos que guarden una determinada relaciÃ³n.

FUNCIONAL SECUENCIAL COMUNICACIONAL	Mayor CohesiÃ³n	MÃ³dulo como caja negra
PROCEDURAL TEMPORAL LÃ GICA COINCIDENTAL	Menor CohesiÃ³n	MÃ³dulo transparente

8.3 TEORÃ A DE LA NORMALIZACIÃ N

◊ Conceptos.
◊ 1FN - Dependencia Funcional.
◊ 2FN - Dependencia Funcional Completa.
◊ 3FN - Dependencia Funcional Transitiva.
◊ Boyce-Codd.

CONCEPTOS GENERALES

◊ **Objetivo**: eliminar la “dependencia” entre los atributos (eliminar la redundancia de datos).
◊ “**NormalizaciÃ³n**”: conjunto de restricciones que deben satisfacer los datos. Cada conjunto de restricciones se conoce como “forma normal”.

1FN - DEPENDENCIA FUNCIONAL

◊ **1FN**: prohÃ–be que en un registro haya grupos repetitivos (“todos los campos han de ser atÃ³micos”). Por tanto, para que un registro estÃ© en 1FN, **no** puede contar con atributos **multivaluados**.

La 1FN estÃ¡ definida formalmente por la Dependencia Funcional.

◊ **Dependencia Funcional**: “Se dice que un descriptor Y (conjunto de campos) depende funcionalmente del descriptor X, si y sÃ³lo si, cada valor de X tiene asociado en todo momento un Ãºnico valor de Y”.	**X Y**

2FN - DEPENDENCIA FUNCIONAL COMPLETA

◊ **2FN**: en el caso de que la clave sea compuesta, un registro estÃ¡ en 2FN “si ademÃ¡s de estar en 1FN,

todos los campos que no formen parte de la clave candidata suministran informaciÃ³n acerca de la clave completa".

◊ **2FN**: "Un registro estÃ¡ en 2FN si estÃ¡ en 1FN y todo campo no clave tiene una dependencia funcional completa respecto de las claves candidatas".

La 2FN estÃ¡ definida formalmente por la Dependencia Funcional Completa.

◊ **Dependencia Funcional Completa**: "Si el descriptor X es compuesto, se dice que Y tiene dependencia funcional completa o plena respecto de X si depende funcionalemente de X pero no depende de ningÃºn subconjunto del mismo".	**X(X1, X2)** **X --> Y** **X1 -/-> Y** **X2 -/-> Y**

3FN - DEPENDENCIA FUNCIONAL TRANSITIVA

◊ **3FN**: para que un registro estÃ© en 3FN, ademÃ¡s de estar en 2FN, los atributos que NO son clave, proporcionan informaciÃ³n SÃ LO sobre la clave (y no acerca de otros atributos).

◊ **3FN**: "Un registro estÃ¡ en 3FN si estÃ¡ en 2FN y ningÃºn campo no clave depende transitivamente de ninguna clave".

La 3FN estÃ¡ definida formalmente por la Dependencia Funcional Transitiva.

◊ **Dependencia Funcional Transitiva**: Dado un registro con tres descriptores en el que existen las siguientes dependencias funcionales: …"se dice que Z tiene una dependencia transitiva respecto de X a travÃ©s de Y".	**X --> Y** **Y --> Z** **Y -/-> X** **X - --> Z**

FORMA NORMAL DE BOYCE-CODD (FNBC)

◊ Se emplea para evitar solapamientos entre claves compuestas.

◊ "Se dice que un registro estÃ¡ en FNBC si, y sÃ³lo si, todo determinante es clave".

♦ **METODOLOGÃ AS DE DISEÃ O DETALLADO DE PROGRAMAS**

◊ Conceptos.

◊ Jackson.

◊ Warnier.

CONCEPTOS GENERALES

- ◊ Permiten obtener una estructura jerÃ¡rquica del programa.
- ◊ Permiten generar una codificaciÃ³n en forma de pseudocÃ³digo (que puede ser fÃ¡cilmente traducida a un lenguaje procedimental de programaciÃ³n).
- ◊ El enfoque que siguen estos mÃ©todos se basa en el anÃ¡lisis de los datos que deben manejar los programas.
- ◊ Toman como punto de partida: una especificaciÃ³n detallada de la entrada, la salida y los algoritmos del programa a construir.
- ◊ Las principales metodologÃ−as para el DiseÃ±o Detallado de Programas son: **el MÃ©todo Jackson** y **la MetodologÃ−a Warnier**.

“POSIBLES” PREGUNTAS CAPÃ TULO 78

1. Las tÃ©cnicas de modelado enfocado en el comportamiento de un SI dependiendo del tiempo:

- · Lista de Eventos.
- · Diagramas de TransiciÃ³n y Redes de Petri.
- · Ambas son correctas.

2. Para efectuar comprobaciones entre los distintos modelos:

- · TÃ©cnicas Matriciales.
- · Historia de Vida de Entidad.
- · Ambas son correctas

3. En anÃ¡lisis, la Historia de vida de Entidades proporciona:

- · Visualizar relaciones existentes entre entidades de un modelo de datos.
- · Una validaciÃ³n para DFD y modelos de datos.
- · Ambas son correctas.

4. Con respecto al AnÃ¡lisis y el DiseÃ±o:

- · AnÃ¡lisis se refiere a QuÃ© hacer y DiseÃ±o a CÃ³mo hacerlo.
- · AnÃ¡lisis se refiere a CÃ³mo hacerlo y DiseÃ±o a QuÃ© hacer.
- · Ninguna es correcta.

5. Un Modelo Conceptual como el modelo E-R:

- · TÃ©cnica utilizada en AnÃ¡lisis.
- · TÃ©cnica utilizada en DiseÃ±o.
- · Ninguna es correcta.

6. El Modelo Relacional es:

- · Modelo Conceptual de Datos.
- · Modelo LÃ³gico de Datos.

· Modelo FÃ–sico de Datos.

7. Las mÃ©tricas que miden la calidad estructural del diseÃ±o:

· VerificaciÃ³n y CohesiÃ³n.
· Integridad y ValidaciÃ³n.
· CohesiÃ³n y Acoplamiento.

8. El grado de independencia entre mÃ³dulos lo mide el atributo de calidad:

· CohesiÃ³n.
· Acoplamiento.
· Ambas son correctas.

9. El Acoplamiento Ã³ptimo:

· Es MÃ–nimo (Acoplamiento Normal).
· Es MÃ¡ximo (Acoplamiento por Contenido).
· Ambas son correctas.

10. La CohesiÃ³n Ã³ptima:

· Es MÃ¡xima (CohesiÃ³n Funcional).
· Es MÃ¡xima (CohesiÃ³n Coincidental).
· Ambas son correctas.

11. La Dependencia Funcional Transitiva define formalmente:

· 1FN.
· 2FN.
· 3FN.

12. Generan una estructura lÃ³gica partiendo de especificaciones detalladas de datos de entrada, de datos de salida y de algoritmos necesarios :

· MetodologÃ–as de DiseÃ±o ArquitectÃ³nico de Programas.
· MetodologÃ–as de DiseÃ±o Detallado de Programas.
· MetodologÃ–as de AnÃ¡lisis Estructurado.

13. Jackson y Warnier:

· Proponen un mÃ©todo para generar una codificaciÃ³n en forma de pseudocÃ³digo.
· Proponen un mÃ©todo para el DiseÃ±o Detallado de Programas.
· Ambas son correctas.

RESPUESTAS

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
c	c	b	a	a	b	c	b	a	a	c	b	c

CAPITULO XII : PRUEBAS DEL SOFTWARE

- **INTRO**
  - ◊ Conceptos.
  - ◊ Definiciones.

CONCEPTOS GENERALES

- ◊ Durante el desarrollo de software se requieren una serie de controles periÃ³dicos para evaluar los productos generados con el fin de encontrar defectos.
- ◊ Las pruebas consisten en *ejecuciones o ensayos*, posteriores a la terminaciÃ³n del cÃ³digo y previos a la entrega del software al cliente.
- ◊ Las pruebas permiten **VERIFICAR** y **VALIDAR** el software:
  - · VERIFICAR: comprobar `si lo estamos haciendo tal como lo habÃ–amos planteado'.
  - · VALIDAR: comprobar `si estamos haciendo lo que nos han pedido'.
- ◊ La verificaciÃ³n y validaciÃ³n se llevan a cabo cuando nuestro producto ya es ejecutable.

DEFINICIONES

- ◊ **Prueba**: proceso de ejecutar un programa con el fin de encontrar errores. (Tb: conjunto de casos y procedimientos de prueba).
- ◊ **Caso de prueba**: entradas, condiciones y resultados preconcebidos para estudiar un objetivo concreto.
- ◊ **Defecto (bug)**: `algo mal programado'. (Ej: variable mal declarada, paso de parÃ¡metros errÃ³neo, etc.).
- ◊ **Fallo**: el sistema (o uno de sus componentes) no satisface los requisitos de rendimiento especificados.
- ◊ **Error**: diferencia entre un valor calculado, observado o medido y e la valor verdadero (“Margen de error”). (Tb: acciÃ³n humana que conduce a un resultado incorrecto. Ej: “el programador pulsa una tecla equivocada” Â¿?).

- **FILOSOFÃ A DE LAS PRUEBAS DEL SOFTWARE**
  - ◊ Conceptos.
  - ◊ Recomendaciones.
  - ◊ Conclusiones.

CONCEPTOS GENERALES

- ◊ **La prueba exhaustiva del software es impracticable** (NO se pueden probar todas las posibilidades de un programa).
- ◊ Descubrir un defecto debe considerarse el “Ã©xito de una prueba”.
- ◊ La realizaciÃ³n de pruebas se trata de una actividad de detecciÃ³n (NO de prevenciÃ³n) de problemas en el software.
- ◊ Los defectos no son siempre el resultado de una negligencia.

RECOMENDACIONES PARA LAS PRUEBAS

- ◊ Definir el resultado de salida esperado y compararlo con el obtenido.
- ◊ El programador debe evitar probar sus propios programas.
- ◊ Inspeccionar a conciencia el resultado de cada prueba.
- ◊ Definir casos de prueba que incluyan datos de entrada tanto vÃ¡lidos y esperados, como no vÃ¡lidos e inesperados.
- ◊ Se debe probar que el software hace lo que debe y NO hace lo que NO debe.
- ◊ Evitar los casos desechables (`casos demasiado lamentables').
- ◊ Asumir que SIEMPRE habrÃ¡ fallos a la hora de hacer planes de prueba.
- ◊ Donde hay un defecto suele haber otros (`Â¿recursividad fallil?').
- ◊ Concienciarnos de que "probar es una tarea tanto o mÃ¡s creativa que el desarrollo de software".

CONCLUSIONES

- ◊ La filosofÃ–a mÃ¡s adecuada para las pruebas consiste en planificarlas y diseÃ±arlas de forma sistemÃ¡tica (para poder detectar el mÃ¡ximo nÃºmero y variedad de defectos con el mÃ–nimo consumo de tiempo y esfuerzo).
- ◊ "Un buen caso de prueba es aquel que tiene una gran probabilidad de encontrar un defecto no descubierto aÃºn".
- ◊ "El Ã©xito de una prueba consiste en detectar un defecto no encontrado antes".

♦ **EL PROCESO DE PRUEBA**

♦ **GeneraciÃ³n del Plan de Pruebas**: en base a la documentaciÃ³n del proyecto y del software a probar.

♦ **DiseÃ±o detallado de pruebas especÃ–ficas**: se especifican los casos y procedimientos (basÃ¡ndose en la documentaciÃ³n del software).

♦ **EjecuciÃ³n de los casos de prueba**: sobre la configuraciÃ³n del software.

♦ **Evaluar los resultados**: comparÃ¡ndolos con las salidas esperadas. La evaluaciÃ³n permite dos actividades:

- ◊ **DepuraciÃ³n** (localizaciÃ³n y correcciÃ³n de defectos): en caso de NO conseguir detectar los defectos puede ser necesario la realizaciÃ³n de pruebas adicionales. Si se corrige un defecto debe volver a probarse el software.
- ◊ **AnÃ¡lisis** de la estadÃ–stica de errores: puede ser Ãºtil para realizar predicciones de la fiabilidad del software y para detectar las causas mÃ¡s habituales de error y mejorar los procesos de desarrollo.
- ◊ **TÃ CNICAS DE DISEÃ O DE CASOS DE PRUEBA**

· Conceptos.
· Enfoques.

CONCEPTOS GENERALES

· **Objetivo**: conseguir una confianza aceptable en la detecciÃ³n de defectos sin necesidad de consumir una cantidad excesivo de recursos. Debe lograrse, por tanto, un equilibrio entre disponibilidad de recursos y confianza en la detecciÃ³n de defectos.
· El diseÃ±o de casos de prueba estÃ¡ totalmente mediatizado por la imposibilidad de probar exhaustivamente el software.
· NO pueden probarse todas las posibilidades del software deben elegirse algunas de ellas que se consideren "representativas". Dificultad: elegir los casos representativos a ejecutar.

PRINCIPALES ENFOQUE PARA EL DISEÃ O DE CASOS

· Estructural.
· Funcional.
· Aleatorio.

· **Enfoque Estructural (o de caja blanca)**: · Se centra en la estructura interna del programa. · Prueba exhaustiva: probar todos los posibles caminos de ejecuciÃ³n.	
· **Enfoque Funcional (o de caja negra)**: · Se centra en la especificaciÃ³n de funciones (entradas y salidas). · Prueba exhaustiva: probar todas las posibles entradas y salidas.	
· **Enfoque Aleatorio**: · Utiliza modelos (normalmente estadÃ−sticos) que representen las posibles entradas al programa. · Prueba exhaustiva: probar todas las posibles entradas.	

· Estos enfoques NO son excluyentes entre sÃ−, ya que se pueden combinar para conseguir una detecciÃ³n de defectos mÃ¡s

eficaz..

◊ **PRUEBAS ESTRUCTURALES**

- · Conceptos.
- · ClasificaciÃ³n de Criterios de Cobertura LÃ³gica.
- · MÃ©trica de McCabe.

CONCEPTOS GENERALES

- · El diseÃ±o de casos tiene que basarse en la elecciÃ³n de caminos importantes que ofrezcan una seguridad aceptable en descubir defectos Para ello se utilizan los **criterios de cobertura lÃ³gica**.
- · No requieren el uso de ninguna representaciÃ³n grÃ¡fica especÃ–fica (aunque es habitual el empleo de los diagramas de flujo de control (*flowcharts*)).

CLASIFICACIÃ_N DE LOS CRITERIOS DE COBERTURA LÃ_GICA

Menor coste

Menor nÂº ejecuciones

◊ Cobertura de sentencias: generar los casos de prueba necesarios para que cada sentencia (instrucciÃ³n) del programa se ejecute al menos vez.

Mayor coste

Mayor nÂº ejecuciones

◊ Cobertura de decisiones: ... cada decisiÃ³n tenga, al menos 1 vez, un resultado verdadero y, al menos 1 vez, un resultado falso.

◊ Cobertura de condiciones: … cada condiciÃ³n de cada decisiÃ³n adopte el valor `verdadero' al menos 1 vez, y el valor `falso' al menos 1 vez.

◊ Criterio de decisiÃ³n/condiciÃ³n: exigir el cumplimiento de la cobertura de decisiones y la cobertura de condiciones.

◊ Criterio de condiciÃ³n mÃºltiple: (en el caso de dividir una decisiÃ³n multicondicional en varias unicondicionales) conseguir que todas las combinaciones posibles

	de resultados (verdadero/falso) de cada condiciÃ³n de cada decisiÃ³n se ejecuten al menos 1 vez. ◊ Cobertura de caminos: …cada camino se ejecute al menos 1 vez.	

Cobertura de Caminos

- Criterio de cobertura mÃ¡s elevado.
- Requiere que cada uno de los caminos del programa se ejecuten al menos 1 vez.
- Camino.definiciÃ³n: “secuencia de sentencias encadenadas desde la sentencia inicial del programa hasta su sentencia final”.
- Los bucles constituyen el elemento que que genera un mayor nÃºmero de problema en la cobertura de caminos.
- Para reducir el nÃºmero de caminos se emplean los **caminos de prueba**: “camino del programa que atraviesa, *como mÃ¡ximo*, 1 vez el interior de cada bucle que encuentra”. (Como alternativa al camino de prueba, otros especialistas recomiendan que se pruebe cada bucle 3 veces: una vez sin entrar en su interior, otra ejecutÃ¡ndolo 1 vez y otra ejecutÃ¡ndolo 2 veces).
- Existen algoritmos basados en matrices (que representan el grafo de flujo del programa) que permiten contar el nÃºmero total de caminos de prueba.

UTILIZACIÃ N DE LA COMPLEJIDAD CICLOMÃ”TICA DE MCCABE

- Objetivo: indicar el nÃºmero de caminos independientes que existen en un grafo.
- By Bruce: a ver… dado el carÃ¡cter teÃ³rico de los presentes resÃºmenes, esta parte me la salto. Si tienes especial interÃ©s en saber cÃ³mo se calcula la complejidad de McCabe a partir de un grafo de flujo, te miras las pÃ¡ginas 402, 403, 404 y 405 del libro. Como adelanto: es otra mariconada mÃ¡s entre tantas que aparecen el el libro. Inciso concluido. Hasta luego .

◊ **PRUEBAS FUNCIONALES**

- · Conceptos.
- · TÃ©cnicas.

CONCEPTOS GENERALES

- · Se centra en el estudio de la especificaciÃ³n del software, del anÃ¡lisis de las funciones que debe realizar, de las entradas y de las salidas.
- · Deben “buscarse criterios que permitan elegir un subconjunto de casos cuya ejecuciÃ³n aporte una cierta confianza en detectar los posibles defectos del software”.
- · **Para elegir los casos de prueba, se considera Caso de Prueba Bien Elegido aquel que**:
- · Reduce el nÃºmero de otros casos necesarios.
- · Cubre un conjunto extenso de otros casos posibles.

TÃ CNICAS DE DISEÃ O DE CASOS DE CAJA NEGRA

- · Particiones o Clases de Equivalencia.
- · AVL.
- · Conjetura de errores.

· **Particiones o Clases de Equivalencia:**

- Cualidades.
- MÃ©todo de DiseÃ±o.

- Cualidades de la TÃ©cnica:
- Cada caso debe cubrir el mÃ¡ximo nÃºmero de entradas.
- Debe tratarse el dominio de valores de entrada (divido en un nÃºmero finito de clases de equivalencia) que cumplan la siguiente propiedad: “la prueba de un valor representativo de una clase permite suponer `razonablemente' que el resultado obtenido serÃ¡ el mismo que el obtenido probando cualquier otro valor de la misma clase.
- MÃ©todo de DiseÃ±o:
- IdentificaciÃ³n de Clases de Equivalencia.
- CreaciÃ³n de Casos de Prueba.
- IdentificaciÃ³n de Clases de Equivalencia:
- IdentificaciÃ³n de las condiciones de entrada (restricciones de formato

o contenido de los datos de entrada).

- IdentificaciÃ³n de las clases de equivalencia (a partir de los datos de entrada). Ã stas clases pueden ser:
  - ◊ De datos vÃ¡lidos.
  - ◊ De datos no vÃ¡lidos o errÃ³neos.
- ♦ “Todos los valores de la clase deben ser tratados de la misma forma por el programa” (Principio de Igualdad de Tratamiento).
- ♦ REGLAS para identificar clases:

Rango de valores	Ej: “El nÃºmero estarÃ¡ comprendido entre 1 y 49”	◊ C. vÃ (1 â nÂ â 49 ◊ C. no vÃ (nÂ < 1 ◊ C. no vÃ (nÂ > 49
NÃºmero de valores	Ej: “se pueden registrar de uno a tres *propietarios* de un piso”	◊ C. vÃ (1 â pro â 3) ◊ C. no vÃ (pr < 1 ◊ C. no vÃ (pro > 3

SituaciÃ³n booleana	Ej: “el primer carÃ¡cter debe ser una letra”	◊ C. vÃ (“e un let ◊ C. no vÃ (“n es un let
Conjunto de valores admitidos (tratando el programa de distinta forma a cada uno de ellos)	Ej: “pueden registrarse tres tipos de inmuebles: pisos, chalÃ©s y locales comerciales”	◊ C. vÃ (“p ◊ C. vÃ (“c ◊ C. vÃ (“l ◊ C. no vÃ (“g
◊ Si se sospecha que ciertos elementos de una clase no se tratan igual que el resto de la misma, deben dividirse en clases menores.		

- ♦ CreaciÃ³n de Casos de Prueba:
- ♦ AsignaciÃ³n de un nÃºmero Ãºnico a cada clase de equivalencia.
- ♦ Hasta que se hayan cubierto todas las clases de equivalencia con casos de prueba escribir un caso que cubra tantas clases vÃ¡lidas como sea posible.
- ♦ Hasta que se hayan cubierto todas las clases de equivalencia no vÃ¡lidas con casos de prueba escribir un caso para una Ãºnica clase no vÃ¡lida sin cubrir.
  - ◊ **AnÃ¡lisis de Valores LÃmite (AVL):**
    - · Conceptos.
    - · Reglas.

- Conceptos Generales:
- Complementa a la tÃ©cnica de particiones de equivalencia.
- Los casos de prueba que exploran las *condiciones lÃmite* de un programa producen mejor resultado para la detecciÃ³n de defectos.
- Condiciones LÃminte.def: “situaciones que se hallan directamente arriba, abajo y en los mÃ¡rgenes de las clases de equivalencia”.
- El proceso de selecciÃ³n de casos es tambiÃ©n heurÃsticoÂ¹.

1. **HeurÃstico:** (gr. ÎµÏ Ï”Î¯Ï€Î·Ï , hallar, inventar), adj. Perteneciente o relativo a la heurÃstica.

- REGLAS para identificar clases:

(1) Rango de valores.

(Como condiciÃ³n de entrada)

Ej: “el valor estarÃ¡ comprendido entre 1 y -1”

- ◊ **2** Casos vÃ¡lidos para los extremos:
- ◊ (- 1.0)
- ◊ (+ 1.0)
- ◊ **2** Casos no vÃ¡lidos para valores justo mÃ¡s allÃ¡ de los extremos:
  - (- 1.001)
  - (+ 1.001)

(2) NÃºmero de valores.

(Como condiciÃ³n de entrada)

Ej: “el fichero de entrada tendrÃ¡ de 1 a 255 registros”

- ◊ MÃ¡ximo: (255 registros)
- ◊ MÃ–nimo: (1 registro)
- ◊ MÃ¡ximo + 1: (256 registros)
- ◊ MÃ–nimo - 1: (0 registros)

(3) Rango de valores.

(Como condiciÃ³n de salida)

Ej: “el descuento mÃ¡ximo aplicable en una compra al contado serÃ¡ el 50%, el mÃ–nimo serÃ¡ el 6%”

- ◊ (Idem regla 1). Se escribirÃ¡n casos para obtener: 6%,

50%, 5.99% y 50.01%.

(4) NÃºmero de valores.

(Como condiciÃ³n de salida)

Ej: “el programa puede mostrar de 1 a 4 listados”

- ◊ (Idem regla 2). Se escribirÃ¡n casos para obtener: 0, 1, 4 y 5 listados.

Conjunto ordenado de elementos.

(Entrada o salida)

Ej: tablas, archivos secuenciales…

- ◊ Caso concentrado en el primer elemento.
- ◊ Caso concentrado en el Ãºltimo elemento.
- ◊ Tanto en la regla 3 como en la 4, debe tenerse en cuenta que:
- ◊ Los valores lÃ−mite de entrada no generan necesariamente los valores lÃ−mite de salida.
- ◊ No siempre se pueden generar resultados fuera del rango de salida.

- ◊ **Conjetura de Errores:**
- ◊ Conceptos.
- ◊ Recomendaciones.

- ◊ Conceptos Generales:
  - · Consiste en:
  - · Enumerar

una lista de *equivocaciones* que pueden cometer los desarrolladores y de *situaciones* propensas a ciertos errores.
- Generar los casos de prueba en base a dicha lista.

♦ La gene de caso se basa en la intui o la expe

♦ Valo prop a erro y Reco a tene en cuen para los caso espe

♦ Valo cero (tant en la salid com en la entra

- En lista o tabla ning valo todo los valo igua
- Reco imag que le prog hubi *inter algo mal* en la espe
- Reco imag lo que el *usua* pued intro com *entra* a un prog

· **PRUEBAS ALEATORIAS**

- Simulan la entrada habitual del programa creando datos de entrada (en la secuencia

y con la frecuencia con las que podrÃ–an aparecer en una situaciÃ³n real), de forma contÃ–nua y sin parar.

- Utilizan una herramienta llamada "**generador de pruebas**", que debe contener la descripciÃ³n de entradas y secuencias de entrada junto con su posibilidad de ocurrir en la prÃ¡ctica.
- Empleadas comunmente en la prueba de

compiladore
- Son menos utilizadas que las tÃ©cnicas de caja blanca y de caja negra.

· **ENFOQUE PRÃ”CTICO RECOMENDADO PARA EL DISEÃ O DE CASOS**
- Pretende mostrar el uso mÃ¡s apropiado de cada tÃ©cnica para la obtenciÃ³n de un conjunto de casos Ãºtiles (sin prejuicio de las estrategias de casos de prueba).

RECOMENDACIO A LA HORA DE DISEÃ AR CASOS

- **Comenzar formando los grafos causa-efecto** si la especificaciÃ contiene combinacion de condiciones de entrada.
- **Usar el anÃ¡lisis de valores-lÃ–**
- **Identificar las clases vÃ¡lidas y no vÃ¡lidas de equivalencia** para la entrada y la salida, y aÃ±adir los casos no incluidos anteriorment
- **Utilizar la tÃ©cnica de conjetura de errores**.
- **Ejecutar los casos generados hasta el**

momento y analizar la cobertura obtenida.
- **Examinar la lÃ³gica del programa**.

· **DOCUMENTACIÃ DEL DISEÃ O DE LAS PRUEBAS**
- Conceptos.
- Documentos
  - ♦ Plan de Prue
  - ♦ Espe del Dise de Prue
  - ♦ Espe de Casc de Prue
  - ♦ Espe de Proc de Prue

CONCEPTOS GENERALES

- Es necesaria tanto para una **mejor organizaciÃ** de las pruebas, como para asegurar

su **reutilizaciÃ**
- Los documentos contemplado en el estÃ¡ndar se asocian a las distintas fases de las pruebas.

DOCUMENTOS

- **Plan de Pruebas:** seÃ±alar el enfoque, los recursos y el esquema de actividades de prueba, asÃ– como los elementos a probar, las caracterÃ–st las actividades de prueba, el personal responsable y los riesgos asociados.

- **Especificaci del DiseÃ±o de Pruebas:** especificar los refinamiento necesarios sobre el enfoque general reflejado en el plan e identificar las caracterÃ–st que se deben probar con este diseÃ±o de pruebas.
- **Especificaci de Casos de Prueba:** definir uno de los casos de prueba identificado por una especificaciÃ del diseÃ±o de las pruebas.
- **Especificaci**

de Procedimien de Prueba: especificar los pasos para la ejecuciÃ³n de un conjunto de casos de prueba.

- **EJECUCIÃ N DE LAS PRUEBAS**
  - Proceso.
  - Documentac
  - DepuraciÃ³n
  - AnÃ¡lisis de Errores.

EL PROCESO DE EJECUCIÃ N

- General.
- EjecuciÃ³n.
- Comprobaci
- **Proceso General de Pruebas:**

- EjecuciÃ³n de las pruebas.
- ComprobaciÃ³n de la TerminaciÃ³n de las Pruebas.
- EvaluaciÃ³n de Resultados.

(NOTA: este subproceso se lleva a cabo en el caso de que las pruebas hayan terminado, en caso contrario hay que generar pruebas adicionales).

- **Subproceso de EjecuciÃ³n:**

· EjecuciÃ³n de las pruebas (`paso por el ordenador de los datos de prueba').
· ComprobaciÃ³n de posibles fallos de ejecuciÃ³n.
· Si ha habido algÃºn fallo se realizan nuevas pruebas.
· Si NO existen fallos, se pasa a la comprobaciÃ³n de la terminaciÃ³n de las pruebas.

- **Subproceso de Comprobac**

· Comprobar si se

cumplen los criterios de compleciÃ³n (descritos en el Plan de Pruebas).
- En caso de terminar las pruebas Evaluar los productos probados sobre la base de los resultados obtenidos.
- En caso de NO terminar las pruebas Comprobar la presencia de condiciones anormales de prueba.
- Si han existido condiciones anormales, se pasa de nuevo a la evaluaciÃ³n. En caso contrario se pasa a generar y ejecutar pruebas adicionales.
    - Los **criterios de compleciÃ³n de pruebas** hacen referencias a Ã¡reas que deben

cubrir las pruebas y el grado de cobertura para cada Ã¡rea (Ej: `se debe cubrir cada caracterÃ–st del software mediante un caso de prueba o una excepciÃ³n aprobada en el plan de pruebas')

DOCUMENTACIÃ DE LA EJECUCIÃ N DE LAS PRUEBAS

DEPURACIÃ N

- Conceptos.
- Consejos.
- **Conceptos Generales:**
  - **Dep** "pro de loca anal y

corr
los
defe
que
se
sosp
que
tiene
el
softv

- Suel
ser
la
cons
de
una
prue
con
Ã©x
- Las
cons
de
una
depu
pued
ser
dos:

- Las
dos
prin
etap
de

la
depu
son:

- **Consejos chorras para la DepuraciÃ³**

LocalizaciÃ³n del error:

- Analizar la informaciÃ³ y pensar.
- Al llegar a un punto muerto, pasar a otra cosa.
- Al llegar a un punto muerto, describir el problema a

otra persona.
- Usar herramientas de depuraciÃ³n sÃ³lo como recurso secundario.
- No experimentar cambiando el programa.
- Se deben atacar los errores individualme
- Se debe fijar la atenciÃ³n tambiÃ©n en los datos.

CorrecciÃ³n del error:

- Donde hay un defecto suele haber mÃ¡s.
- Debe fijarse el defecto, no sus sÃ–ntomas.
- La probabilidad de corregir perfectamen

un defecto no es del 100%.
- Cuidado con crear nuevos defectos.
- La correcciÃ³n debe situarnos temporalmen en la fase de diseÃ±o.
- Cambiar el cÃ³digo fuente, no el cÃ³digo objeto.

ANÃ”LISIS DE ERRORES (o AnÃ¡lisis Causal)

- **Se emplea para obtener informaciÃ³ sobre la "naturaleza de los defectos".**
- El objetivo principal es

la formaciÃ³n del personal sobre los errores que comete para que se puedan prevenir en el futuro.

12.11 ESTRATEGIA DE APLICACIÃ N DE LAS PRUEBAS

- Conceptos.
- Prueba de Unidad.
- Prueba de IntegraciÃ³n
- Prueba del Sistema.
- Prueba de AceptaciÃ³n

CONCEPTOS GENERALES

- Conceptos.
- Niveles de Prueba.
- Niveles de Prueba - Productos de Desarrollo.
- **Conceptos:**

- Pretende integrar el diseÃ±o de los casos de prueba en una serie de pasos bien coordinados (a travÃ©s de la creaciÃ³n de distintos niveles de prueba con diferentes objetivos).
- Permite la coordinaciÃ³ del personal de desarrollo, del departament de aseguramien de calidad y del cliente (gracias a la definiciÃ³n de los papeles que

deben desempeÃ±a cada uno de ellos y la forma de llevarlos a cabo).

- **Etapas de la Estrategia de Pruebas (“Niveles de Prueba”):**

· Prueba de MÃ³dulo (prueba de unidad): el propio personal de desearrollo (normalmente) realiza la prueba de cada mÃ³dulo.
· Prueba de IntegraciÃ³n: se integran los mÃ³dulos probados (en base al esquema del diseÃ±o del software), para comprobar sus interfaces en el trabajo conjunto.
· Prueba de ValidaciÃ³n

(o funcional): el software totalmente ensamblado se prueba como un conjunto para comprobar si cumple o no los requisitos.

- Prueba del Sistema: se integra el software ya validado con el resto del sistema para probar su funcionamiento conjunto.
- Prueba de AceptaciÃ³n: el usuario comprueba el producto final en su propio entorno de explotaciÃ³n para aceptarlo como estÃ¡, o no.
    - **RelaciÃ³n entre Niveles de Prueba y Etapas de Desarrollo:**
    - La relaciÃ³n entre productos de desarrollo

y niveles de prueba se concreta en un modelo de ciclo de vida denominado **“Ciclo en Uve”**.

- Prueba de MÃ³dulo (prueba de unidad): ejercitar la *lÃ³gica del mÃ³dulo* (caja blanca) y la *especificaciÃ³n de las funciones* que debe realizar el mÃ³dulo (caja negra) “EspecificaciÃ³n y lÃ³gica de mÃ³dulo”.
- Prueba de IntegraciÃ³n: tratar los *mecanismos de agrupaciÃ³n de mÃ³dulos* (fijados en la estructura del programa) “DiseÃ±o modular”.
- Prueba de ValidaciÃ³n (o

funcional): comprobar posibles *desajustes* entre el software y los requisitos fijados para su funcionamiento en la ERS (No asociado a ningÃºn producto del desarrollo segÃºn el `Ciclo en Uve').

- Prueba del Sistema: comprobaciones del *cumplimiento de objetivos* indicados para el sistema "EspecificaciÃ³n de requisitos".
- Prueba de AceptaciÃ³n: para que el *usuario verifique si el producto final* se ajusta a los requistos por Ã©l fijados (o al contrato) "Requisitos de usuario".

PRUEBA DE UNIDAD

- Conceptos.
- Definiciones
- Enfoque.
- **Conceptos Generales:**
- Suelen ser realizadas por el personal de desarrollo (evitando que las realice el programador
- Se intenta el mÃ¡ximo acercamiento al ideal de exhaustivida (a travÃ©s de los criterios de cobertura lÃ³gica y funcional).
- La planificaciÃ³ de este nivel de pruebas se realiza para la globalidad de pruebas

de unidad.

- Consisten en las pruebas formales que nos permiten declarar que un mÃ³dulo estÃ¡ listo y terminado.
- Puede realizarse tanto a un Ãºnico mÃ³dulo como a un conjunto indefinido de Ã©stos.
- **Definiciones**

- "**MÃ³dulo.d** bloque bÃ¡sico de construcciÃ³ de programas.
- "**MÃ³dulo.d** parte del cÃ³digo que implementa una funciÃ³n simple.
- "**MÃ³dulo.d** fragmento de cÃ³digo

que se puede compilar independient

- “**Unidad de prueba**”: uno o mÃ¡s mÃ³dulos que cumplen las siguientes condiciones:
  - ♦ Todo pert al mis prog
  - ♦ Al men uno de ellos NO ha sido prob
  - ♦ El conj de mÃ³ es el obje de un proc de prue
- **Enfoque para una Prueba de Unidad:**
- Orientado claramente al

diseÃ±o de casos de **caja blanca** (aunque complement con caja negra).
- Suele seguirse el `Enfoque Recomendad
- Las razones por las que incluir casos de caja blanca son:
    - El *tama* del mÃ³ (o grup de mÃ³ es *apro* para pode usar tÃ© de caja blan para exar toda la lÃ³g
    - El *tipo de*

defe
que
se
busc
coin
con
los
prop
de
la
lÃ³g
deta
de
mÃ³

PRUEBA DE INTEGRACIÃ_N

- Conceptos.
- IntegraciÃ³n Incremental.
  - ♦ Asce
  - ♦ Desc
- IntegraciÃ³n NO Incremental.
- Sandwich.
- Incremental - No Incremental.
- Incremental ascendente - Incremental descendente.
- Recomendac
- **Conceptos Generales:**
- **Objetivo**: probar las interfaces (`flujos de datos') entre componentes o mÃ³dulos.
- Consiste en

integrar los distintos componentes del software (`realiaziÃ³n de pruebas de los distintos mÃ³dulos'), hasta contar con el producto global (`sistema completo').

- Los tipos fundamentales de integraciÃ³n son: la **integraciÃ³n incremental** (ascendente y descendente)y la **integraciÃ³n no incremental** (o `Big-Bang').
- NO es siempre obligatorio que, en la estrategia de aplicaciÃ³n de las

pruebas, deba terminarse la prueba de todos los mÃ³dulos antes de empezar a integrar. (“La prueba de mÃ³dulos y las pruebas de integraciÃ³n se solapan”).

- El orden de integraciÃ³n elegido influye en aspectos como:
  - La form de prep caso
  - Las herra nece
  - El orde de codi y prob los mÃ³
  - El coste

de
la
depu
- El
coste
de
la
prep
de
caso

- **IntegraciÃ³n Incremental Ascendente:**
- Comenzando por los mÃ³dulos de mÃ¡s bajo nivel (`mÃ³dulos hoja'), se combina el siguiente mÃ³dulo que se debe probar con el conjunto de mÃ³dulos que ya estÃ¡n probados. Se incrementa progresivam el nÃºmero de mÃ³dulos hasta formar el programa

completo.

- Fases: (En el ejemplo se parte de un esquema idÃ©ntico al de la fase 4).

· Se combinan los mÃ³dulos de bajo nivel en grupos (no obligatoriamente) que realizan alguna subfunciÃ³n especÃ–fica. (Para reducir el nÃºmero de pasos de integraciÃ³n).

· Se escribe para cada grupo un mÃ³dulo *impulsor*.

· Se prueba cada grupo probando su impulsor.

· Se eliminan los impulsores de cada grupo y se sustituyen

por los mÃ³dulos del nivel superior de la jerarquÃ–a.

- En todas las integracione incrementale se combinan pruebas de unidad y de integraciÃ³n
- **IntegraciÃ³ Incrementa Descendent**
- Comenzando por el mÃ³dulo de control principal (`mÃ³dulo raÃ–z'), se van incorporando mÃ³dulos subordinado progresivam hasta formar el programa completo.
- Se basa en el empleo de *modulos ficticios*, cuya codificaciÃ³

es mÃ¡s compleja que la creaciÃ³n de impulsores.

- Deb
  a la dific de cons exist vari nive de sofis de mÃ³ ficti (de may a men nive de com
- MÃ³ que sÃ³l mue un men de traza
- MÃ³ que mue los parÃ que se les pasa
- MÃ³ que devu valo que NO

depe
de
los
parÃ
de
entra

- MÃ³
que
devu
valo
que
depe
de
los
parÃ
de
entra

- Se escri mÃ³ ficti para simu la pres de los subo ause que serÃ llam por el mÃ³ raÃ-
- Se susti uno de los subo

ficti
por
el
mÃ³
corr
segÃ
el
orde
eleg
y
se
inco
ficti
para
reco
las
llam
del
Ãºlti
inco

- Se
ejecu
las
corr
prue
cada
vez
que
se
inco
un
mÃ³
nuev
- Al
term
cada
prue
se
susti
el
ficti
por
su
corr
real.
- [Cor
repe
algu
caso
de
prue
de

ejecu
ante
para
aseg
de
que
no
se
ha
inclu
ning
defe
nuev

- Se prue cada mÃ³ por sepa
- Se ensa todo los mÃ³ para form el prog com
- Se prue el conj

- Se com en los mÃ³ de nive supe de form desc
- Se com simu de

form
desc
desd
los
mÃ³
infer

- Se
cont
hasta
que
amb
apro
se
encu
en
algÃ
nive
inter

Inte Incr

PRU
DEI
SIS

PRU
DE
ACI

“PO
PRE
CAR
XII

1.
Obje
de
TÃ©
para
Dise
de
Casc
de
Prue

2.
Los

tres
enfo
prin
para
el
Dise
de
Casc
de
Prue

3.
Las
Prue
Estru
se
basa
el
la
elec
de
cam
que
ofrez
una
segu
acep
a
la
hora
de
desc
defe
utiliz
los
llam

4. La Prue Func o Caja Neg se cent en:

5. Clas de Equi

6. El Enfo PrÃ¡ reco

7. El Info de Inci

8. La Dep

9. La Prue Beta se reali en el ento real de traba

10. Dad
la
espe
El
nom
es
de
6
cara

11. Dad
el
sigu
diag
mod
supo
que
no
hay
ning
tipo
de
agru
de
mÃ³
para
hace
prue
de

unid

RES

1	**2**
c	c

ME' ES FÃ" (V 2.0)

Intr

Com com que salv un par de ellas toda las

preg
apar
en
el
exan
parc
del
terce
trim
aunq
la
may
no
de
form
litera

En
la
parte
dedi
a
las
preg
del
test
de
MÃ
lo
que
se
han
indic
han
sido
las
RES
COR
(NO
son
disti
opci
entre
las
que
haya
que
esco
AsÃ
que
si
leem

por
ejem

**4.
Â¿C
son
los
prin
proc
de
un
proc
de
anÃ
estru**

-
Mod
de
proc

-
Mod
conc
de
dato

-
Mod
lÃ³g
de
dato

-
Espe
de
inter
de
usua

Vien
a
deci
que
los
prod
del
proc
de
anÃ
estru
son

esos
cuat
Los
mem
Y
si
en
el
exan
te
encu
con
una
opci
que
no
se
corr
con
ning
de
esos
cuat
es
porq
dich
opci
es
falsa

La
preg
11
del
Ãºlti
exan
la
6
del
apar
Con
del
Siste
de
Infor
de
los
“apu
aquÃ
pres
diga
que.

mmr
esto
CAS
Nos
opta
por
la
resp
que
se
inclu
en
el
test
de
MÃ
que
es
tamb
la
que
se
expo
aquÃ
(que
tamb
es
la
que
se
dio
por
vÃ¡l
en
el
ante
exan
y
que,
muy
curio
cont
a
lo
que
pode
leer
en
el
apar
dedi
a

la
activ
CSI
5
de
MÃ©
A
cont
con
todo
nosc
la
suso
preg
junto
a
su
corre
resp

**6.
Â¿Q
part
NO
inte
en
la
activ
CSI
5
“Eje
de
las
Pru
del
Siste**

-
TÃ©
de
Siste

-
TÃ©
de
Com

-
Equi
del
Proy

-
Ana

Y
si
nos
loca
el
cuad
resu
de
la
activ
CSI
5,
nos
enco
con
lo
sigu

Tare
CSI 5.1
CSI 5.2
CSI 5.3

Lo
que
vien
a
deci
que
nos

apre
las
resp
a
mod
de
dogr
de
fe
y
que
la
suer
estÃ
pres
dura
el
exar

Proc
Prin

MÃ
3
es
una
met
orie
a
proc
que
prete
abar
el
ciclo
de
vida
com
del
softv
Por
cons
tene
que,
los
proc
prin
de
su
estru
son
la

plan
el
desa
y
el
man
del
SI.

El
proc
de
desa
del
SI
a
su
vez
se
desc
en
otros
5
proc
4
de
los
cual
serÃ
conv
mem
en
el
caso
de
prete
apro
la
parte
dedi
a
MÃ
del
exan

Proc
que
hay
que
estu

Obte
una
espe
deta
del
SI
que
se
plas
en
un
catÃ
de
requ

Los
prod
gene
en
este
proc
son:
el
catÃ
de
requ
el
glos
y
el
cont
del
siste

Defi
de
la
arqu
del
SI,
del
ento
tecn
que
le
va

a
dar
sopo
y
la
espe
deta
de
sus
com

Se
gene
las
espe
de
cons
la
desc
tÃ©
del
plan
de
prue
la
defi
de
los
requ
de
impl
y
el
dise
de
los
proc
de
migr
y
carg
inici

Gen
el
cÃ³
de

los
com
del
SI,
desa
los
proc
de
oper
y
segu
y
elab
los
man
de
usua
final

Entr
y
acep
del
siste
en
su
total
y
reali
de
toda
las
activ
nece
para
el
paso
a
prod

AN
DEL
SIS
DE
INF

1. Obj del proc de AnÃ de Siste de Info (ASI

- La obte de una espe deta del siste que satis las nece de infor de los usua y sirva de base para el post disE del siste

2. QuÃ tipo de desa cont MÃ 3:

- Estru

-
Orie
a
obje

**3.
La
tÃ©
de
Cas
de
Uso
se
utili
en
el:**

-
AnÃ
orie
a
obje

-
AnÃ
estru

**4.
Â¿C
son
los
prin
proc
de
un
proc
de
anÃ
estru**

-
Mod
de
proc

-
Mod
conc
de
dato

-

Mod
lÃ³g
de
dato

-
Espe
de
inter
de
usua

**DIS
DEI
SIS
DE
INF**

**1.
Obj
del
proc
de
Dise
de
Siste
de
Info
(DSI**

-
Defi
la
arqu
del
SI,
el
ento
tecn
que
le
va
a
dar
sopo
y
la
espe
deta
de
sus
com

2. Los proc gene en el DSI son:

- Espe de Con

- Desc TÃ© del Plan de Prue

- Defi de los Requ de Impl

- Dise de los Proc de Mig y Carg Inici (si proc

**3. Los obje que se pers con la activ

**“Dis
de
la
Arq
de
MÃ³
del
Siste
(DS
5)
son:**

-
Real
el
dise
deta
de
la
inter
de
usua

-
Con
las
inter
entre
mÃ³
de
cada
subs
entre
subs
y
con
el
resto
de
siste

-
Defi
los
mÃ³
del
siste
de
infor
iden
los
proc

que se van a impl en cada subs espe

4. Las tare que hay que lleva a cabo para cum los obje de la acti DSI 7 "Ve y acep de la arqu del siste son:

- Veri de la calid tÃ© de cada mod o espe

- Aseg

la
cohe
entre
los
disti
mod

-
Acep
del
diseA
de
la
arqu
por
parte
de
los
resp
de
Expl
y
Siste

**5.
Los
obje
a
cum
de
la
activ
Espe
TÃ©
del
Plan
de
Prue
(DSI
10)
son:**

-
Dete
los
Con
o
Apti
adic
que
requ
los

usua
final
para
oper
con
el
nuev
siste

-
Com
la
Plan
de
las
Prue
(dete
los
disti
perfi
impl
en
su
prep
ejecu
y
eval
de
resu

-
Dise
Deta
de
los
Nive
de
Prue
espe
en
el
proc
de
AnÃ
del
Siste
de
Info
(ASI

-
Defi

con
prec
las
verif
a
reali
sobr
el
siste
conf
a
los
nive
de
prue
estab

**CON
DEL
SIST
DE
INF**

**1.
Obj
del
proc
de
Con
del
Sist
de
Info
(CSI**

-
Gen
cÃ³c
de
com
del
SI.

-
Elab
los
proc
de
oper
y
segu

-
Elab
man
de
usua
final

**2.
Tare
de
la
activ
CSI
1
“Pro
del
Ento
de
Gen
y
Con**

-
Impl
de
la
Base
de
Dato
FÃ–
o
Fich

-
Prep
del
ento
de
Con

**3.
La
gene
del
cÃ³d
corr
a
cada
uno
de
los
com**

del siste de info se reali a part de la info del proc

- Dise del Siste de Info (DSI

4. Obj de las Pru Unit

- Com que la estru de cada uno de los com del Siste de Info es la corr y que se ajust a

la
func
estab

**5.
Obj
de
las
Pru
de
Inte**

-
Veri
si
los
com
o
subs
inter
corr
a
trav
de
sus
inter
inter

-
Veri
si
los
com
o
subs
inter
corr
a
trav
de
sus
inter
exte

-
Veri
si
los
com
o
subs
cubr

la
func
estab

-
Veri
si
los
com
o
subs
se
ajust
a
los
requ
espe

**6.
Â¿Q
part
NO
inte
en
la
activ
CSI
5
“Eje
de
las
Pru
del
Siste**

-
TÃ©
de
Siste

-
TÃ©
de
com

-
Equi
del
proy

-
Anal

7. SegÃ la activ CSI 8 “Co de los Con y Proc de Mig y Carg inici de Datc Â¿q acci es nece ante de lleva a cabo la gene de cÃ³c

- Prep de la infra tecn nece para reali la codi

IMP Y ACI DEL SIST

1.

Obj
prin
del
proc
de
Imp
y
Ace
del
Siste
(IAS

-
Entr
y
acep
del
siste
en
su
total
para
reali
el
paso
a
prod

2.
En
la
activ
IAS
“Est
del
plan
de
impl

-
Se
revis
la
estra
estab
en
el
proc
Estu
de
Viab
del

Siste

3. Aspe cons dent del Plan de Forr del Equ de Imp en la activ IAS 2 “For Nec para la Imp

- Esqu de form

- Mate de form

- Recu de form

- Plan de la form

4. Proc obte en la activ

IAS 4 "Ca de dato al Ente de Ope

- Base de dato / Fich carg

5. Part de las activ de Pru de Imp y Pru de Ace del SI:

- Jefes de Proy

- Resp de impl

- Usua expe

- Dire de los

usua

6. Â¿Q afirm son cierto sobr la figu del Resp de Mar del siste

- Form parte del equi de impl

- Reci una form adec para adqu una visiÁ glob del siste

- Anal en detal el siste que va a impl

- Valo si

la
infor
dispo
es
sufic
para
abor
el
futur
man
del
siste

-
Llev
a
cabo
las
prue
de
impl

**7.
Obj
de
la
activ
IAS
8
"Est
de
Acu
de
Nive
de
Serv**

-
Dete
los
servi
que
requ
el
siste

-
Espe
los
nive
de
servi

con los que se va a valo la calid del servi

- Defi los com que se adqu con la entre del siste

8. Una vez efec las pru de impl y acep com paso prev a la pues en proc la apro del SI es forn por:

-

Com
de
Dire

**ME
ES
FÃ”
(V
2.0)**

Intr

Com
com
que
salv
un
par
de
ellas
toda
las
preg
apar
en
el
exar
parc
del
terce
trim
aunq
la
may
no
de
form
litera

En
la
parte
dedi
a
las
preg
del
test
de
MÃ
lo
que

se
han
indic
han
sido
las
RES
COR
(NO
son
disti
opci
entre
las
que
haya
que
esco
AsÃ
que
si
leem
por
ejem

**4.
Â¿C
son
los
prin
proc
de
un
proc
de
anÃ
estru**

-
Mod
de
proc

-
Mod
conc
de
dato

-
Mod
lÃ³g

de
dato

-
Espe
de
inter
de
usua

Vier
a
deci
que
los
prod
del
proc
de
anÃ
estru
son
esos
cuat
Los
mem
Y
si
en
el
exan
te
encu
con
una
opci
que
no
se
corr
con
ning
de
esos
cuat
es
porq
dich
opci
es
falsa

La preg 11 del Ãºlti exan la 6 del apar Cons del Siste de Infor de los “apu aquÍ pres diga que. mmr esto CAS Nose opta por la resp que se inclu en el test de MÃ que es tamb la que se expo aquÍ (que tamb es la que se

dio
por
vÃ¡l
en
el
ante
exar
y
que,
muy
curio
cont
a
lo
que
pode
leer
en
el
apar
dedi
a
la
activ
CSI
5
de
MÃ
A
cont
con
todo
nosc
la
suso
preg
junto
a
su
corre
resp

**6.
Â¿C
part
NO
inte
en
la
activ
CSI
5**

“Eje
de
las
Pru
del
Siste

-
TÃ©
de
Siste

-
TÃ©
de
Com

-
Equi
del
Proy

-
Ana

Y
si
nos
loca
el
cuad
resu
de
la
activ
CSI
5,
nos
enco
con
lo
sigu

Tare
CSI 5.1

CSI 5.2	
CSI 5.3	

Lo que vien a deci que nos apre las resp a mod de dogr de fe y que la suer estÃ pres dura el exar

Proc Prin

MÃ 3 es una meto orie a proc

que
prete
abar
el
ciclo
de
vida
com
del
softv
Por
cons
tene
que,
los
proc
princ
de
su
estru
son
la
plan
el
desa
y
el
man
del
SI.

El
proc
de
desa
del
SI
a
su
vez
se
desc
en
otros
5
proc
4
de
los
cual
serÃ
conv

mem
en
el
caso
de
prete
apro
la
parte
dedi
a
MÃ
del
exan

Proc
que
hay
que
estu

Obte
una
espe
deta
del
SI
que
se
plas
en
un
catÃ
de
requ

Los
prod
gene
en
este
proc
son:
el
catÃ
de
requ
el
glos
y

el
cont
del
siste

Defi
de
la
arqu
del
SI,
del
ento
tecn
que
le
va
a
dar
sopo
y
la
espe
deta
de
sus
com

Se
gene
las
espe
de
cons
la
desc
tÃ©
del
plan
de
prue
la
defi
de
los
requ
de

impl
y
el
dise
de
los
proc
de
migr
y
carg
inici

Gen
el
cÃ³d
de
los
com
del
SI,
desa
los
proc
de
oper
y
segu
y
elab
los
man
de
usua
final

Entr
y
acep
del

siste
en
su
total
y
reali
de
toda
las
activ
nece
para
el
paso
a
prod

**ANA
DEL
SIST
DE
INF**

**1.
Obj
del
proc
de
AnÃ
de
Sist
de
Info
(ASI**

-
La
obte
de
una
espe
deta
del
siste
que
satis
las
nece
de
infor
de
los

usua
y
sirva
de
base
para
el
poste
diseÃ
del
siste

**2.
QuÃ
tipo
de
desa
cont
MÃ©
3:**

-
Estru

-
Orie
a
obje

**3.
La
tÃ©
de
Cas
de
Uso
se
utili
en
el:**

-
AnÃ
orien
a
obje

-
AnÃ
estru

4.

Â¿C
son
los
prin
proc
de
un
proc
de
anÃ
estru

-
Mod
de
proc

-
Mod
conc
de
dato

-
Mod
lÃ³g
de
dato

-
Espe
de
inter
de
usua

DIS
DEL
SIST
DE
INF

1.
Obj
del
proc
de
Dise
de
Siste
de
Info

(DSI

- Defi
la
arqu
del
SI,
el
ento
tecn
que
le
va
a
dar
sopo
y
la
espe
deta
de
sus
com

**2.
Los
proc
gene
en
el
DSI
son:**

- Espe
de
Con

- Desc
TÃ©
del
Plan
de
Prue

- Defi
de
los
Requ

de
Impl

-
Dise
de
los
Proc
de
Mig
y
Carg
Inici
(si
proc

**3.
Los
obje
que
se
pers
con
la
activ
“Dis
de
la
Arq
de
MÃ
del
Siste
(DS
5)
son:**

-
Real
el
dise
deta
de
la
inter
de
usua

-
Con
las
inter

entre mÃ³ de cada subs entre subs y con el resto de siste

- Defi los mÃ³ del siste de infor iden los proc que se van a impl en cada subs espe

4. Las tare que hay que lleva a cabo para cum los obje de la activ DSI

7 “Ve y acep de la arqu del siste son:

- Veri de la calid tÃ© de cada mod o espe

- Aseg la cohe entre los disti mod

- Acep del diseÃ de la arqu por parte de los resp de Expl y Siste

5. Los obje

a cum de la activ Espe TÃ© del Plan de Prue (DSI 10) son:

- Dete los Con o Apti adic que requ los usua final para oper con el nuev siste

- Com la Plan de las Prue (dete los disti perfi impl en su prep ejecu y eval

de
resu

-
Dise
Deta
de
los
Nive
de
Prue
espe
en
el
proc
de
AnÃ
del
Siste
de
Infor
(ASI

-
Defi
con
prec
las
verif
a
reali
sobr
el
siste
conf
a
los
nive
de
prue
estab

**CON
DEL
SIST
DE
INF**

**1.
Obj
del
proc**

de Con del Sist de Info (CSI

- Gen cÃ³c de com del SI.

- Elab los proc de oper y segu

- Elab man de usua final

2. Tare de la activ CSI 1 “Pro del Ento de Gen y Con

- Impl de la Base

de
Dato
FÃ–
o
Fich

-
Prep
del
ento
de
Con

**3.
La
gene
del
cÃ³d
corr
a
cada
uno
de
los
com
del
siste
de
info
se
reali
a
part
de
la
info
del
proc**

-
Dise
del
Siste
de
Info
(DSI

**4.
Obj
de
las
Pru**

Unit

- Com
que
la
estru
de
cada
uno
de
los
com
del
Siste
de
Infor
es
la
corre
y
que
se
ajust
a
la
func
estab

**5.
Obj
de
las
Pru
de
Inte**

- Veri
si
los
com
o
subs
inter
corre
a
trav
de
sus
inter
inter

- Veri
si
los
com
o
subs
inter
corre
a
trav
de
sus
inter
exte

- Veri
si
los
com
o
subs
cubr
la
func
estab

- Veri
si
los
com
o
subs
se
ajust
a
los
requ
espe

**6.
Â¿Q
part
NO
inte
en
la
activ
CSI
5**

“Eje
de
las
Pru
del
Siste

-
TÃ©
de
Siste

-
TÃ©
de
com

-
Equi
del
proy

-
Anal

7.
Segú
la
activ
CSI
8
“Co
de
los
Con
y
Proc
de
Mig
y
Carg
inici
de
Dato
Â¿q
acci
es
nece
ante
de
lleva
a

cabo la gene de cÃ³c

- Prep de la infra tecn nece para reali la codi

IMP Y ACE DEL SIST

1. Obj prin del proc de Imp y Ace del Siste (IAS

- Entr y acep del siste en su total para reali el paso a prod

2. En la activ IAS "Est del plan de impl

- Se revis la estra estab en el proc Estu de Viab del Siste

3. Aspe cons dent del Plan de Forr del Equ de Imp en la activ IAS 2 "For Nece para la Imp

- Esqu de

form

- Mate de form

- Recu de form

- Plan de la form

4. Proc obte en la activ IAS 4 "Ca de dato al Ento de Ope

- Base de dato / Fich carg

5. Part de las activ de Pru de Imp y

Pru de Ace del SI:

- Jefes de Proy

- Resp de impl

- Usua expe

- Dire de los usua

6. Â¿Q afirm son ciert sobr la figu del Resp de Mar del siste

- Forn parte del equi de impl

- Reci una

form
adec
para
adqu
una
visiÁ
glob
del
siste

-
Ana
en
deta
el
siste
que
va
a
impl

-
Valo
si
la
infor
disp
es
sufic
para
abor
el
futur
man
del
siste

-
Llev
a
cabo
las
prue
de
impl

**7.
Obj
de
la
activ
IAS**

8 “Est de Acu de Nive de Serv

- Dete los servi que requ el siste

- Espe los nive de servi con los que se va a valo la calid del servi

- Defi los com que se adqu con la entre del siste

8. Una vez

efec
las
pru
de
imp
y
acep
com
pas
prev
a
la
pue
en
pro
la
apr
del
SI
es
forn
por:

-
Com
de
Dire

“PO
PRE
MÃ
3.0

1.
La
estru
de
MÃ
V3:

2. MÃ
V3
divi
los
proc
prin
en:

3. Obje
del
Proc
de
Dise
del
Siste
de
Info
(DS

4.
Los
prod
gene
en
el
proc
DSI:

5. Los obje pers en la activ DSI 5 “Dis de la arqu de mÃ³ del Siste

6. Las tarea que hay que lleva a cabo para cum los obje de la activ DSI 7 “Ver y Acep de la Arqu del Siste

7. Los obje de la activ DSI 10 “Esp TÃ© del Plan de Prue

8. Obje
del
proc
de
Con
del
Siste
de
Info
(CSI

♦ Las
tarea
de
la
activ
CSI
1
“Pre
del
Ento

de
Gen
y
Con