

WUOLAH



TEAM_GETPPID__
www.wuolah.com/student/TEAM_GETPPID__



24545

POSIBLES_PREGUNTAS_EXAMEN.pdf

Preguntas que caen en los exámenes (Estudia esto y aprobado seguro)



2º Ingeniería del Software



Grado en Ingeniería Informática



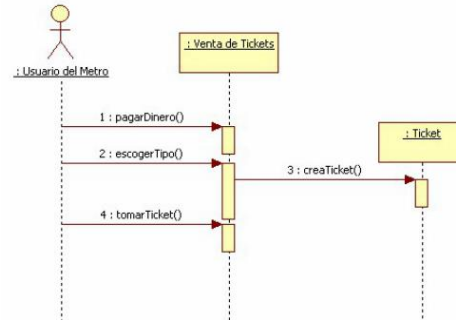
**Escuela Politécnica Superior de Córdoba
UCO - Universidad de Córdoba**

1) ¿QUÉ DIAGRAMAS SE USAN PARA LOS ASPECTOS DINÁMICOS DEL SISTEMA? *

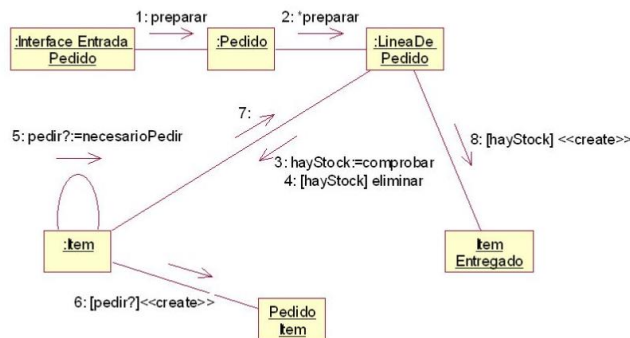
Son 3 los diagramas usados para modelar el aspecto dinámico del sistema:

1) **Diagrama de Interacción:** Describen la interacción entre objetos. Existen 2 tipos de diagrama de interacción.

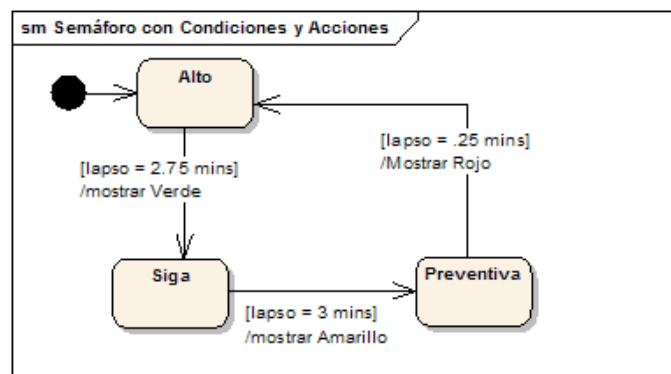
- **Diagrama de Secuencia:** Este diagrama se destacan la ordenación temporal de los eventos. Gráficamente son una table que representa objetos dispuestos a lo largo del eje X y mensajes ordenados por tiempo en el eje Y.



- **Diagrama de Colaboración:** Este diagrama se destacan la organización de los objetos que participan en una interacción y se representa mediante un grafo donde los nodos son los objetos y los arcos los enlaces.



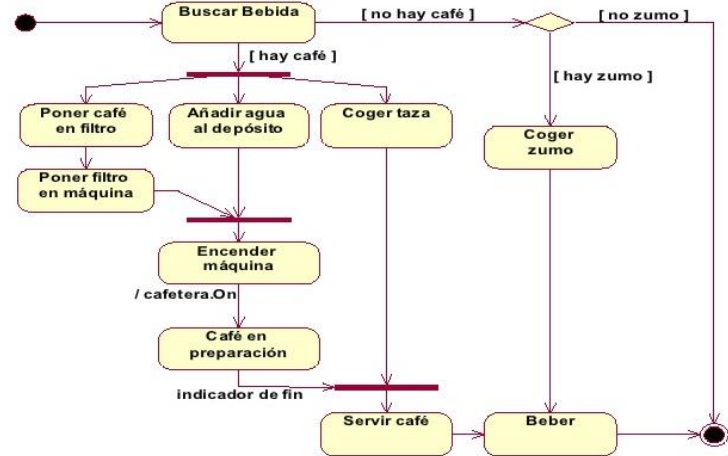
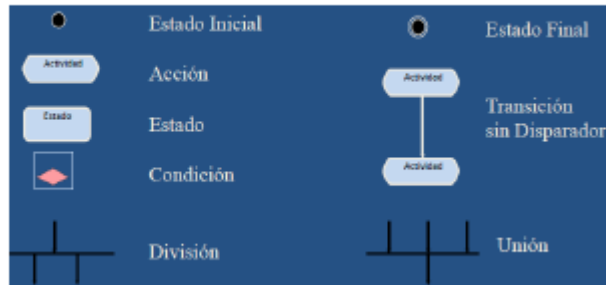
2) **Diagrama de Estado:** Explican el comportamiento de un sistema. Que explican todos los estados posibles por los que puede pasar un objeto en particular y la manera en que la modifica el estado del objeto, así como el resultado de los eventos que llegan a él.



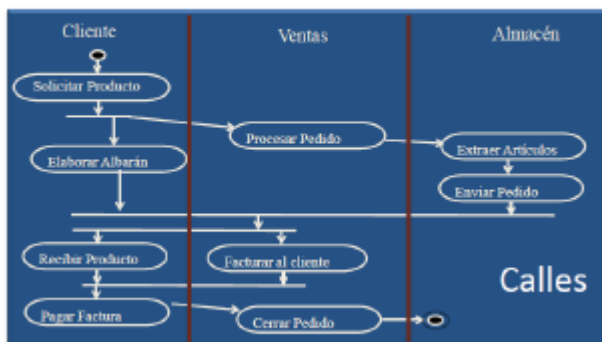
- 3) **Diagrama de Actividades:** El diagrama de actividades es una variación directa del **diagrama de estados**, únicamente que el diagrama de actividades está enfocado a las actividades y a los eventos que hacen cambiar de actividad y que no precisamente los nodos son actividades sino también pueden ser estados.

Por ejemplo, en un diagrama de estados, los nodos son: encendido, funcionando, reiniciado, apagado, como lo ven son estados mientras que en un diagrama de actividades los nodos son encender, poner en funcionamiento, reiniciar o apagar, como lo notan son actividades que implican una acción.

ELEMENTOS BASICOS



Calles o Swimlanes



Flujo de Trabajo



2) EN LOS DIAGRAMAS DE ESTADO, INDIQUE QUE ES UN EVENTO, QUE EVENTOS EXISTEN Y DE SU CLASIFICACION POR TIPO.

Evento: Especificación en UML de un acontecimiento significativo a lo largo del tiempo. En el contexto de máquina de estado modelan los estímulos.

Los eventos pueden ser:

o **Síncronos (1 Subtipo):**

- 1) Llamadas o invocación de operaciones.

o **Asíncronos (3 Subtipos):**

- 1) Señales
- 2) Paso del tiempo
- 3) Cambio de estado que puedan ocurrir en cualquier instante.

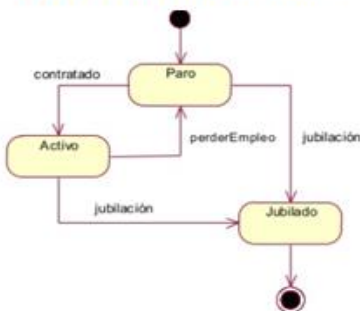
Los eventos son de tipo:

o **Externo:** Entre el sistema y sus actores.

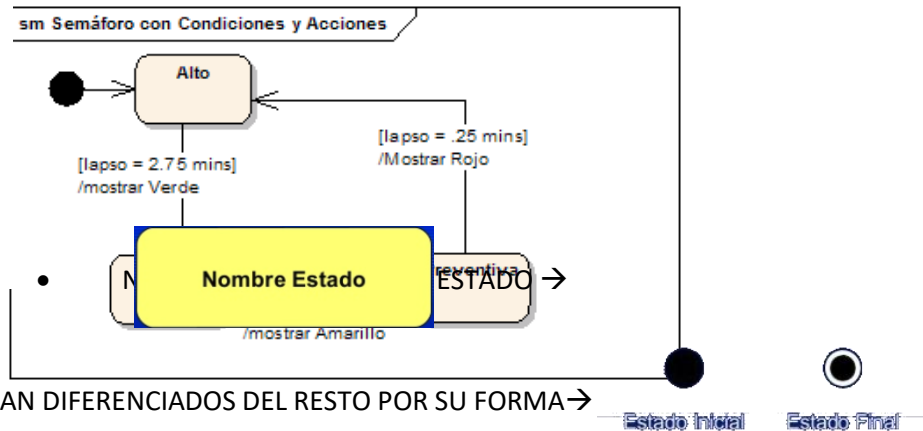
o **Interno:** Entre los objetos del sistema.

3) MODELA UN DIAGRAMA DE ESTADOS ESPECIFICANDO UNA DESCRIPCIÓN NO MUY DETALLADA DE COMO SE REPRESENTA

DIAGRAMA DE ESTADOS DEL OBJETO PERSONA

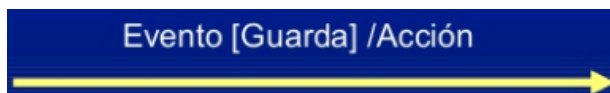


1. Cada objeto está en un estado en un cierto instante.
2. El estado va caracterizado parcialmente por los valores de los atributos del objeto.
3. El estado en el que se encuentra el objeto caracteriza sus condiciones dinámicas.



- ESTADOS INICIAL Y FINAL ESTAN DIFERENCIADOS DEL RESTO POR SU FORMA →
- UNA TRANSICION PUEDE CONTENER OPCIONALMENTE LOS SIGUIENTES ELEMENTOS:
 - **EVENTO:** Acontecimiento significativo a lo largo del tiempo. En el contexto de máquina de estado modelan los estímulos. Los eventos pueden tener argumentos y la transición entre estados es instantánea, esto se debe a la ocurrencia de eventos.
 - **GUARDA:** Permiten condicionar la transición y son condiciones lógicas que retornan **true** o **false**. LA transición solo ocurre solo si la guarda se resuelve como **true**.
 - **ACCION:** Consecuencia de una transición.

NOTACION →



4) INDICA QUE ES UN DIAGRAMA CASOS DE USO Y ESPECIFICA SUS TIPOS SEGÚN SU CLASIFICACION

Un **DIAGRAMA DE CASO DE USO** especifica el comportamiento de un sistema sin tener que especificar como se implementa dicho comportamiento, y es una descripción de unas secuencias o acciones que ejecuta el sistema para producir un resultado, tienen un equilibrio entre genérico y específico. Pueden servir de herramienta de validación del sistema.

El **DIAGRAMA DE CASO DE USO** de uso permite obtener una comprensión del sistema a los desarrolladores, usuarios finales...

TIPOS DE DIAGRAMAS DE CASOS DE USO

- **Esenciales o de trazo grueso:** En ellos se ignoran detalles sobre la forma de interacción entre el actor y el sistema. Sólo se incluyen las alternativas más relevantes y no se entra en detalle sobre las acciones que realiza el sistema.
- **Implementación o trazo fino:** Completan todos los detalles que no se han especificado anteriormente, completamos las alternativas de todo tipo y especifica con más detalle el comportamiento interno del sistema.
- **Temporales:** Aquellos que su inicio queda determinado por el paso del tiempo.
- **Primarios:** Corresponden a los procesos de negocio.
- **Secundarios:** Son necesarios para que el sistema funcione con normalidad.



15%
DE DESCUENTO
WWW.SABWAY.ES



15%
DE DESCUENTO
WWW.SABWAY.ES



15%
DE DESCUENTO
WWW.SABWAY.ES

5) TECNICAS A CONSIDERAR PLANO-TIEMPO (METODOLOGIAS ESTRUCTURADAS)

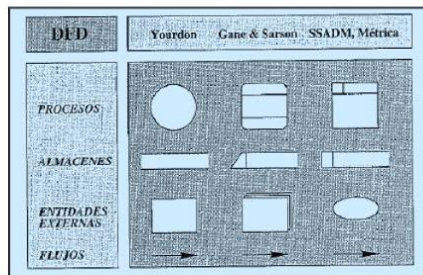
Existen dos técnicas:

- Plano información-tiempo:** Se usan los **Diagramas de la Historia de vida de las entidades (HVE)**, que muestra el efecto del tiempo sobre unas entidades de datos. También se usan las matrices **Entidad-Evento** que especifican las relaciones entre las entidades e interrelaciones de un diagrama E-R y un conjunto de eventos.
- Plano función-tiempo:** Redes Petri y DIAGRAMAS de Transición de Estados, que permiten mostrar el efecto del tiempo sobre un conjunto de funciones del sistema.

6) QUE ES UN DFD (Diagrama de flujo de datos) Y QUE INCLUYE SU ESTRUCTURA.

Un DFD (diagrama de flujo de datos) es una representación en forma de red que refleja el flujo de la información y las transformaciones que se aplican sobre ella al moverse desde la entrada hasta la salida del sistema y se utiliza para modelar las funciones y los datos del sistema a distintos niveles de abstracción

ESTRUCTURA SEGÚN LAS DIFERENTES METODOLOGIAS:



PROCESOS

- Representan un componente que transforma los flujos de datos de entrada en uno o varios flujos de salida
- NO es un programa en ejecución, sino función que debe de realizar un sistema.
- El flujo de salida es igual al de entrada+alguna información. "La regla de conservación de datos"

ALMACENES DE DATOS

- Representan información del sistema almacenada de forma temporal, representando datos que se encuentran "en reposo"
- Se trata de dispositivos lógicos de almacenamiento

Entidades Externas

- Representan un generador o consumidor de información del sistema y que no pertenece al mismo
- Puede representar un subsistema, una persona, departamento, organización, etc., que proporcione datos al sistema

Flujo de datos

- Se define como "Un camino a través del cual viajan datos de composición conocida de una parte del sistema a otra"
- Se representan por arcos dirigidos
- Según su persistencia en el tiempo pueden ser discretos o continuos

7) QUE ES UN DIAGRAMA DE ESTRUCTURA

Diagrama de las Técnicas estructuradas. Es un diagrama en forma de árbol con los siguientes elementos:

- Módulo:** unidad de diseño que representa una división de software clara y manejable con sus interfaces definidas. Puede representar un programa, subprograma o rutina
- Modulo predefinido:** modulo disponible en la biblioteca del sistema
- Conexión:** representa una llamada de un módulo a otro
- Parámetro:** información que se intercambian los módulos: de datos y de control
- Almacén de datos:** lugar donde están almacenados los datos
- Dispositivo físico:** representación de cualquier dispositivo por el que se envíe o reciba información que necesite el sistema

8) TABULAR LA CLASIFICACION DE LAS PRINCIPALES TECNICAS DE MODELADO EN METODOLOGIA ESTRUCTURADAS.

CLASIFICACION SEGÚN SU ESPECIFICACIÓN:

TABLA QUE PREGUNTA Nº1

| | Información | Función | Tiempo |
|-------------|---|---------|--------|
| Información | <ul style="list-style-type: none"> • Especificación de los tipos de entidad • Especificación de los tipos de interrelación • Especificación de los tipos abstractos de datos TAD | | |
| Función | <ul style="list-style-type: none"> • Diccionario de datos • Especificación de los procesos • Especificación de entidades Externas | | |
| Tiempo | <ul style="list-style-type: none"> • Definición de funciones • Especificación de Eventos | | |

ESPECIFICACIÓN

CLASIFICACION SEGÚN SU MODELADO:

TABLA QUE PREGUNTA Nº2

| | Información | Función | Tiempo |
|-------------|---|---------|--------|
| Información | <ul style="list-style-type: none"> • Diagrama Entidad Interrelación • Diagrama de estructura de datos • Matriz Entidad/Entidad | | |
| Función | <ul style="list-style-type: none"> • Diagrama Flujo de datos • Matriz Entidad/Función • Diagrama de flujo de datos • Diagrama descomposición funcional • Diagrama estructural. | | |
| Tiempo | <ul style="list-style-type: none"> • Diagrama Historia Vida de Entidad • Matriz Entidad/Evento • Diagrama de transición de estados. • Lista de eventos • Diagrama de transición de estados | | |

MODELADO

9) JERARQUIA DE CONTROL(Diseño->Conceptos fundamentales del diseño) *

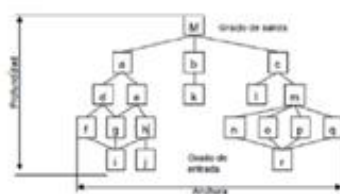
Es uno de los 8 conceptos fundamentales del diseño y es:

La jerarquía de control representa dos características de la arquitectura del programa:

- La visibilidad indica el conjunto de componentes que pueden ser invocados o sus datos ser utilizados por un componente dado (incluso indirectamente).
- La conectividad indica el conjunto de componentes a los que directamente se invoca o sus datos son utilizados en un determinado módulo.

Jerarquía de Control (cont.)

- La profundidad es el nº de niveles de control
- La anchura es la amplitud global del control
- El grado de salida es el nº de módulos controlados por otros módulos
- El grado de entrada es el nº de módulos que controlan directamente a un módulo dado



10) CONCEPTOS FUNDAMENTALES DEL DISEÑO



ABSTRACCION

Tipos a considerar:

- **FUNCIONAL:** Subprogramas parametrizados.
- **DE DATOS:** Especificar datos u objetos.
- **DE CONTROL**

REFINAMIENTO

Es un proceso de elaboración que permite ampliar una declaración original, dando cada vez más detalles.

MODULARIDAD

Atributo del software que permite que sea intelectualmente manejable al estar dividido en componentes que se integran para satisfacer los requisitos □ Sistema modular = Σ módulos

PROCEDIMIENTOS SOFTWARE

Se centra en los detalles de procesamiento de cada módulo individualmente

OCULTACION DE LA INFORMACION

Capacidad de un componente para que la información (procedimiento y datos) contenida dentro del mismo sea inaccesible a otros componentes que no la necesitan.

CONCURRENCIA

Los sistemas software pueden ser clasificados en secuenciales y concurrentes. En unos solo una porción esta activa. En otros los procesos son independientes y activados de forma simultánea.

En los sistemas concurrentes existen problemas específicos como: Bloqueo/desbloqueo, exclusión mutua y sincronización:

VERIFICACION

Un diseño es verificable si puede demostrarse que el diseño generará el producto que satisface los requisitos del cliente.

ESTETICA

Un producto estéticamente agradable es fácilmente reconocido

11) ASPECTOS A CONSIDERAR EN EL DISEÑO MODULAR

Un diseño modular reduce la complejidad y da una vista más fácil de implementación en el desarrollo en paralelo del sistema. La abstracción y la ocultación de la información se usan para definir módulos dentro de una arquitectura del software.

Se deben de considerar:

1. QUE TIPO DE MODULO ES:

- 1.1. **Módulos secuenciales:** Se referencia y ejecuta sin interrupción aparente. Son los que se presentan más frecuentemente.
- 1.2. **Módulos incrementales:** Puede ser interrumpido antes de la terminación por software de la aplicación.
- 1.3. **Módulos paralelos:** Se ejecuta simultáneamente con otro módulo en entornos de multiprocesadores paralelos.

2. LAS CARACTERISTICAS DEL CONJUNTO MODULAR

3. INDEPENDENCIA FUNCIONAL ENTRE MODULOS(COHESION Y ACOPLAMIENTO) **ESTO SUELE PREGUNTARLO EN UNA UNICA PREGUNTA(11.1:Independencia funcional del diseño modular), HABRA QUE EXPLICAR LA COHESION Y EL ACOPLAMIENTO.**

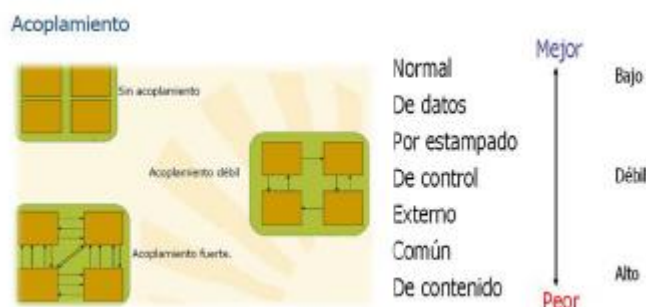
Se consigue desarrollando módulos con una función única y una disminución de la interacción entre los módulos. Se mide usando dos criterios cualitativos:

- **COHESION (MORE IS BETTER):** Es una medida de la fuerza relativa funcional de un módulo.

El objetivo de la cohesión es diseñar servicios robustos y altamente cohesionados cuyos elementos estén fuerte y genuinamente relacionados entre sí.



- **ACOPLAMIENTO (LOWER IS BETTER):** Es una medida de la interdependencia relativa entre los módulos. Un bajo acoplamiento implicará que un cambio en un componente no implicará un cambio en otro.



El objetivo es conseguir el acoplamiento lo mas bajo posible, que nos indica un sistema bien dividido y puede conseguirse mediante la eliminación o reducción de relaciones innecesarias

TIPOS DE ACOPLAMIENTO:

- Acoplamiento normal (N. bajo)
- Acoplamiento de datos (N. bajo)
- Acoplamiento por estampado (N. bajo)
- Acoplamiento de control (N. Medio)
- Acoplamiento externo (N. Medio)
- Acoplamiento común (N. Alto)
- Acoplamiento de contenido (N. Alto)



FELIZ NAVIDAD

15%

DE DESCUENTO

WWW.SABWAY.ES



FELIZ NAVIDAD

15%

DE DESCUENTO

WWW.SABWAY.ES



FELIZ NAVIDAD

15%

DE DESCUENTO

WWW.SABWAY.ES

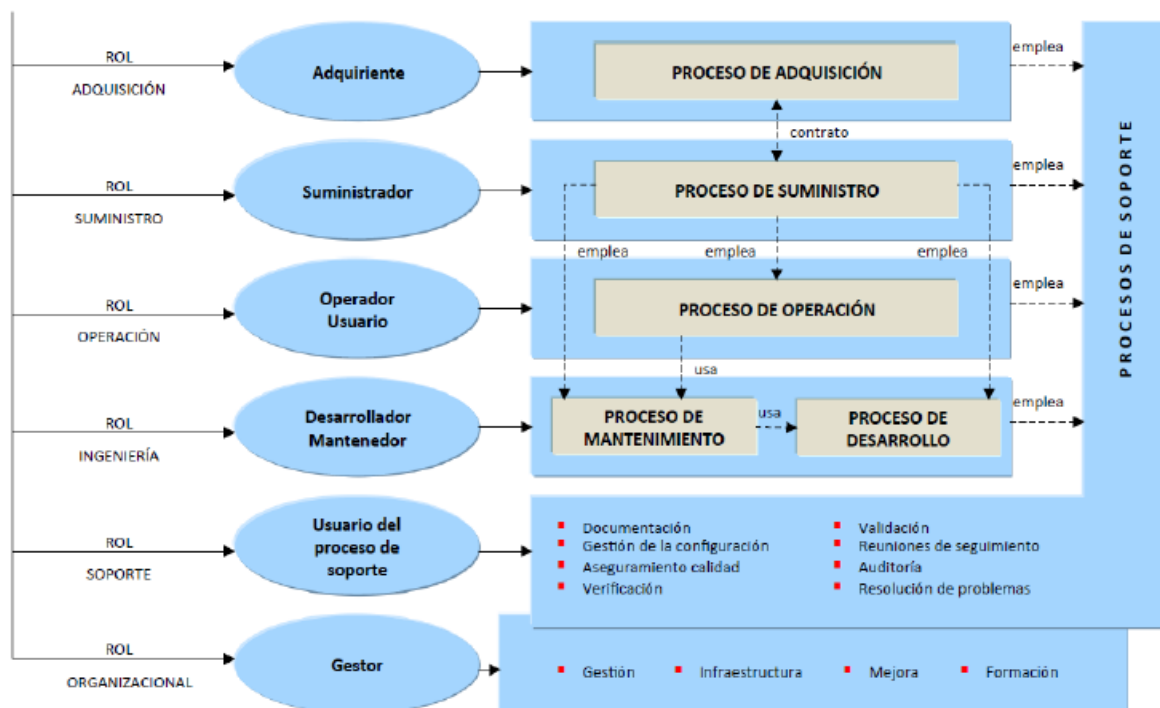
12) REPRESENTA/ESTABLECE EL PROCESO DE PRUEBAS DE MANERA GRAFICA

Le cascas el siguiente esquema y a correr:



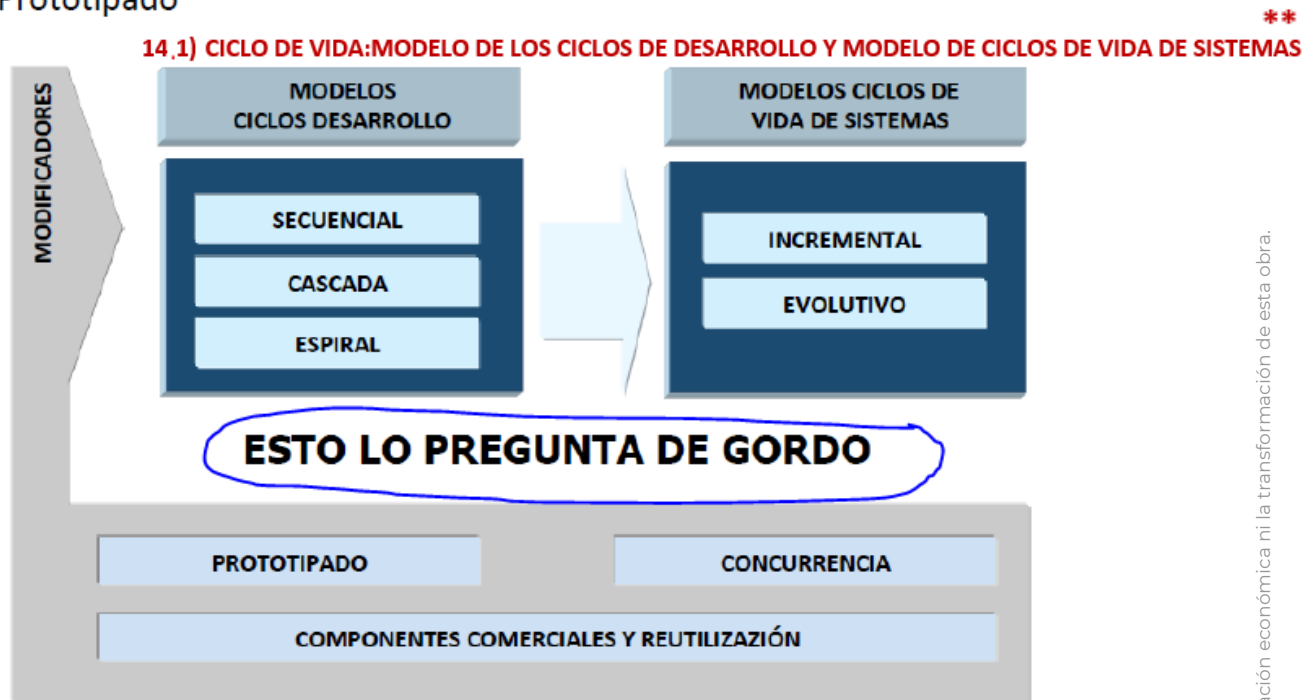
13) VISION GENERAL DE LOS PROCESOS, RELATIONS Y ROLES (TEMA 2- El proceso de desarrollo)

VISIÓN GENERAL DE LOS PROCESOS, RELATIONS Y ROLES



14) PARADIGMAS O MODELOS DE DESARROLLO *

- Conceptos básicos de partida definidos en el estándar ISO 12207
 - *Ciclo de vida*: Periodo desde los requisitos hasta el fin del uso.
 - *Procesos*: Actividades y mecanismos implicados en el ciclo de vida.
- En la etapa del desarrollo, patrones básicos:
 - Secuencial
 - Desarrollo Cascada
 - Desarrollo Espiral
- Una vez desarrollada una primera versión, su ciclo de vida discurre según un patrón:
 - Desarrollo incremental.
 - Desarrollo evolutivo.
- Sobre estos patrones básicos, pueden incluirse modificadores como:
 - Concurrencia
 - Componentes comerciales
 - Prototipado



Modelos de ciclos de evolución

INCREMENTAL

- Este modelo mitiga la rigidez del modelo en cascada, descomponiendo el sistema en partes; en las cuales se aplica un ciclo de desarrollo.
- Las ventajas que ofrecen son:
 1. Usuario tiene al alcance subsistemas que ayudan a perfilar las necesidades.
 2. Este modelo permite entregas parciales en periodos cortos de tiempo.

EVOLUTIVO

- Modelo formado por varios ciclos de desarrollo, cada uno produce un sistema completo
- Es un ciclo de vida común a todos los sistemas desarrollo que se mejoran en versiones sucesivas.

15) DIFERENCIAS ENTRE PERT Y CPM

Ambos son usados en los métodos para la planificación temporal:

PERT (Diagrama de Flechas): PERT es básicamente un método para analizar las tareas involucradas en completar un proyecto dado, especialmente el tiempo para completar cada tarea, e identificar el tiempo mínimo necesario para completar el proyecto total.

CPM (Diagrama de Precedencia o Actividades): es una herramienta para la programación de actividades en la planeación de un proyecto. Es un método de construcción de un diagrama de red del cronograma del proyecto que utiliza cajas, denominados nodos, para representar las actividades y los conecta con flechas que muestran las dependencias.

Diferencias entre PERT y CPM las principales son:

- 1- PERT es más utilizado en ingenierías clásicas y no de software.
- 2- Son representaciones duales Perl representa los nodos como nodos de inicio y fin y en CPM son actividades.
- 3- Conexión entre nodos PERL inicio de tareas y CPM tiempo entre tareas.

16) DIAGRAMA DE GANNT

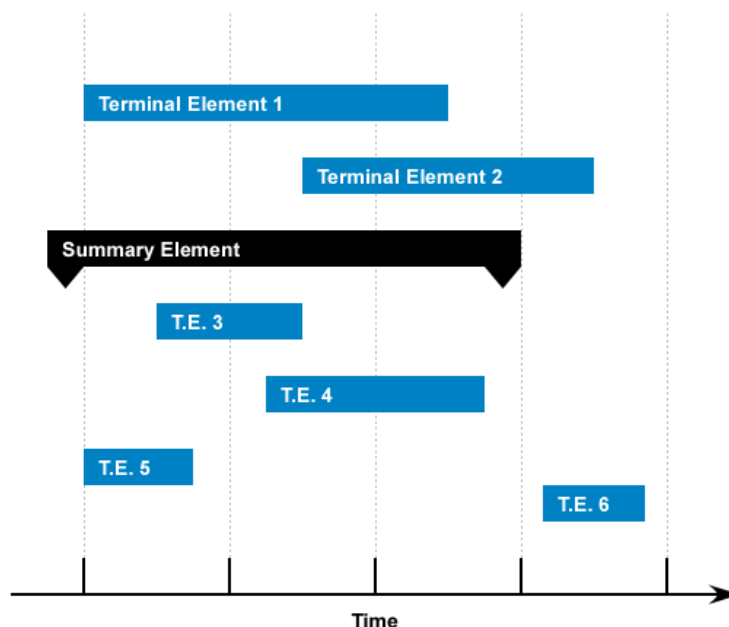
Métodos en la Planificación Temporal

Existen varios métodos para representar de modo visual la secuencia de tareas:

- Diagrama de Gantt
- Diagrama de precedencias
- Diagrama de flechas.

Es un método de la planificación temporal de un proyecto que sirve para representar de manera grafica la secuencia de tareas del proyecto.

El diagrama de Gantt es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. A pesar de esto, el diagrama de Gantt no indica las relaciones existentes entre actividades.



17) ESPECIFICACION DE REQUISITOS Y SUS TECNICAS *

Un **requisito** describe los servicios que debe proporcionar el sistema y sus restricciones. Las principales actividades que componen la especificación de requisitos son:

- 1 **Extracción de requisitos:** Proceso por el cual los clientes descubren, revelan y comprenden los requisitos que desean.
- 2 **Análisis de requisitos:** Proceso de razonamiento sobre los requisitos obtenidos en la etapa anterior, detectando y resolviendo posibles inconsistencias o conflictos.
- 3 **Especificación de requisitos:** Proceso de redacción o registro de los requisitos. Para este proceso puede recurrirse al lenguaje natural, formal, modelos, gráficos etc.
- 4 **Validación de requisitos:** Proceso de confirmación por parte de los usuarios/clientes de los requisitos especificados en el que se comprueba su validez, consistencia, compleción (completitud) etc.

Las principales técnicas de captura y análisis de requisitos son:

- **Entrevistas**
- **Desarrollo conjunto de aplicaciones**
- **Prototipado**
- **Observación**

18) PROCESO DE ANALISIS DE REQUISITOS DE LOS CASOS DE USO *

Un **caso de uso** es un **diagrama** que describe las principales secuencias de iteración entre el sistema y los actores externos.

El proceso de análisis se describe en este orden:

1. Identificar actores
2. Identificar casos de cada actor
3. Identificar nuevos casos de usuario a partir de los existentes
4. Crear descripciones de casos de uso de trazo grueso
5. Definir prioridades y seleccionar pasos de 1ª Iteración
6. Escribir los casos de trazo fino y crear prototipos de interfaces

El mecanismo de ejecución de un caso de uso sigue generalmente 4 fases:

- El actor envía al sistema una petición.
- El sistema valida la petición y los datos.
- EL sistema altera su estado interno.
- El sistema devuelve el resultado al actor.

19) DIAGRAMA DE PRECEDENCIAS *

Es un método de la planificación temporal de un proyecto que sirve para representar de manera gráfica la secuencia de tareas del proyecto:

- Es un **grafo ordenado** donde las tareas/actividades se representan como nodos y las relaciones entre tareas son las flechas. (**El diagrama más usado aparte del de Gantt**)
- Las fechas van de la tarea antecesora a la sucesora.
- Se pueden establecer relaciones de tipo:
 - FIN a COMIENZO
 - COMIENZO a FIN
 - COMIENZO a COMIENZO
 - FIN a FIN



TAREA/ACTIVIDAD: Parte del proyecto que se lleva a cabo durante un periodo de tiempo.

HITO: Punto en el tiempo que marca el inicio o fin de una **TAREA/ACTIVIDAD**.

Las **TAREA/ACTIVIDAD/NODOS** son especificados como:

| Etiqueta actividad | | Duración | |
|--------------------------|-----------------------------|----------------|--------------|
| Inicio temprano | DESCRIPCIÓN DE LA ACTIVIDAD | Final temprano | Final tardío |
| Inicio tardío | | Final tardío | |
| Máximo tiempo disponible | | Holgura | |

- Descripción de la actividad: nombre dado a la actividad
- Etiqueta actividad: número que identifica a cada actividad
- Duración: tiempo que calculamos que se tardará en completar la tarea
- Inicio temprano: fecha más temprana en que puede comenzar la tarea
- Final temprano: fecha más temprana en que puede finalizar la tarea
- Inicio tardío: fecha más retrasada en la que se puede comenzar sin que afecte a la fecha de terminación del proyecto
- Final tardío: fecha más retrasada en la que puede terminar la tarea sin afecte a la fecha final del proyecto
- Máximo tiempo disponible: tiempo máximo que puede durar una tarea en caso de comenzar en su Inicio temprano y concluir en su Final tardío
- Holgura: tiempo que disponemos para jugar con el inicio de la tarea, sin afectar al proyecto

20) DIAGRAMA DE SECUENCIA Y COMUNICACIÓN/COLABORACION *

Los diagramas de SECUENCIA y COMUNICACIÓN/COLABORACIÓN son dos tipos de diagramas que pertenecen a los diagramas de interacción, recordamos que los diagramas de interacción se usan para modelar el aspecto dinámico del sistema (**PREGUNTA Nº1**).

Diagrama de Interacción: Describen la interacción entre objetos. Existen 2 tipos de diagrama de interacción.

- **Diagrama de Secuencia:** Este diagrama se destacan la ordenación temporal de los eventos. Gráficamente son una table que representa objetos dispuestos a lo largo del eje X y mensajes ordenados por tiempo en el eje Y.

Respecto a los diagramas de **COMUNICACIÓN/COLABORACION** tiene 2 características que los distinguen:

- **LÍNEA DE VIDA:** Línea vertical que representa a un objeto a lo largo del tiempo.
- **EL FOCO DE CONTROL:** Rectángulo que representa el periodo de tiempo que el objeto ejecuta la acción.

- **Diagrama de Colaboración:** Este diagrama se destacan la organización de los objetos que participan en una interacción y se representa mediante un grafo donde los nodos son los objetos y los arcos los enlaces.

Respecto a los diagramas de **SECUENCIA** tiene 2 características que los distinguen:

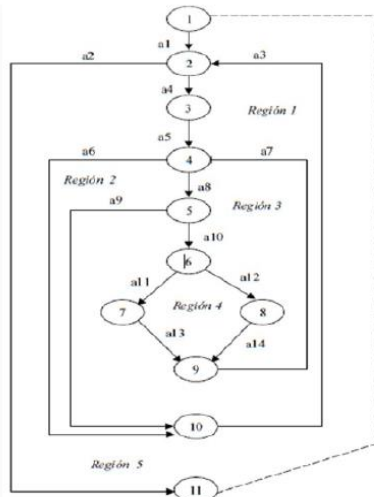
- **EL CAMINO:** Se usan los estereotipos (local, global, self)
- **EL NUMERO DE SECUENCIA:** Indica la ordenación de los mensajes y se utiliza la notación decimal Dewey (1, 1.1, 1.1.1)

21) QUE ES UN DFC (Diagrama de flujo de control) *

Los **DFC** son una extensión de los **diagramas de flujo de datos** (**PREGUNTA N°6**) usados como base en las PRUEBAS ESTRUCTURALES O DE CAJA BLANCA.

Este diagrama permite reflejar el **flujo de control y el procesamiento de control**; muestran como fluyen los sucesos entre los distintos procesos e ilustran como los sucesos externos hacen que se activen los procesos. Muy usado para sacar la **complejidad ciclomatica**.

El DFC contiene los mismos procesos que el DFD, pero muestra el flujo de control en lugar de datos.



- a) $V(G) = 14 - 11 + 2 = 5$
- b) $V(G) = 5$ regiones cerradas
- c) $V(G) = 5$ condiciones

$V(G) \rightarrow$ Complejidad ciclomatica: es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. A mayor valor más riesgo.

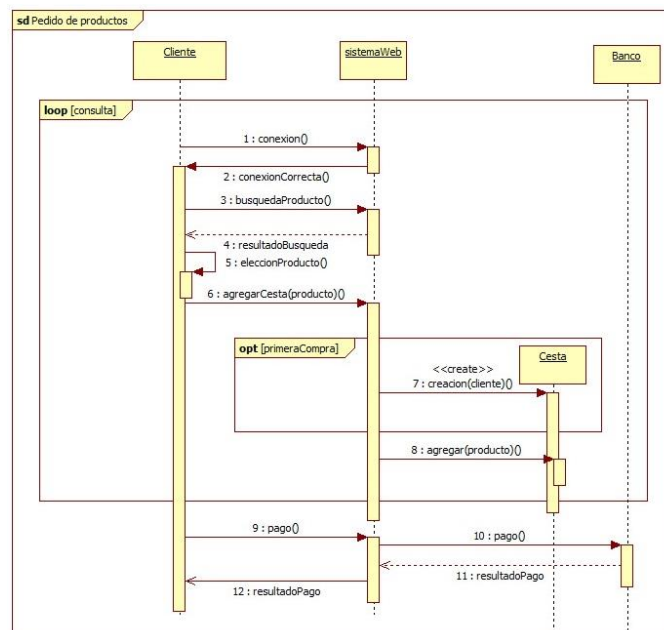
El resultado obtenido en el cálculo de la complejidad ciclomatica define el número de caminos independientes dentro de un fragmento de código

| Complejidad Ciclomática | Evaluación del Riesgo |
|-------------------------|--|
| 1-10 | Programa Simple, sin mucho riesgo |
| 11-20 | Más complejo, riesgo moderado |
| 21-50 | Complejo, Programa de alto riesgo |
| 50 | Programa no testeable, Muy alto riesgo |

22) DESCRIBE LOS OPERADORES DE CONTROL QUE SE USAN EN UML 2.X PARA LOS DIAGRAMAS DE SECUENCIA.

Un marco de interacción es una parte del diagrama de secuencia asociado a una etiqueta, la cual contiene un **operador de control** que determina el modo de ejecución de esa secuencia. A continuación, veremos las principales modalidades de ejecución.

- **Ejecución opcional(opt):** Se trata de una alternativa, que se obtiene utilizando el operador opt seguido de una condición de test. El marco se ejecuta si dicha condición es cumplida.
-
- **Ejecución iterativa o bucle(loop):** El marco se ejecuta mientras se cumple una condición, se efectuará mediante el operador loop, seguidos de los parámetros opciones **min** y **max**



23) DESCRIBE LOS TIPOS DE MODELOS EN EL DISEÑO MODULAR Y SUS CARACTERISTICAS

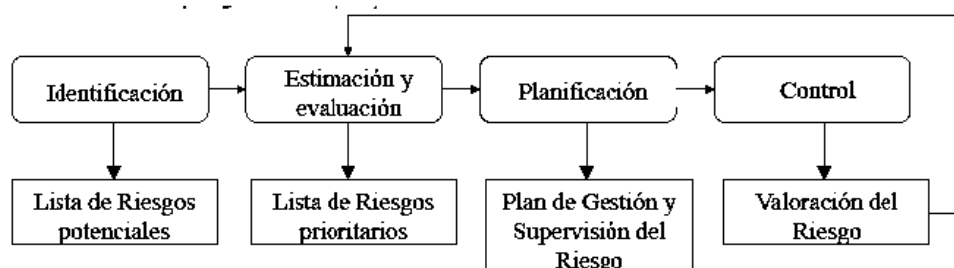
TIPO DE MODULO:

- **Módulos secuenciales:** Se referencia y ejecuta sin interrupción aparente. Son los que se presentan más frecuentemente. **Macros en tiempo de compilación, subprogramas....**
- **Módulos incrementales:** Puede ser interrumpido antes de la terminación por software de la aplicación. Mantiene un puntero que permite al módulo restablecer el punto de interrupción. **Módulos útiles en sistemas con interrupciones.**
- **Módulos paralelos:** Se ejecuta simultáneamente con otro módulo en entornos de multiprocesadores paralelos. **Se encuentran en cálculos de alta velocidad.**

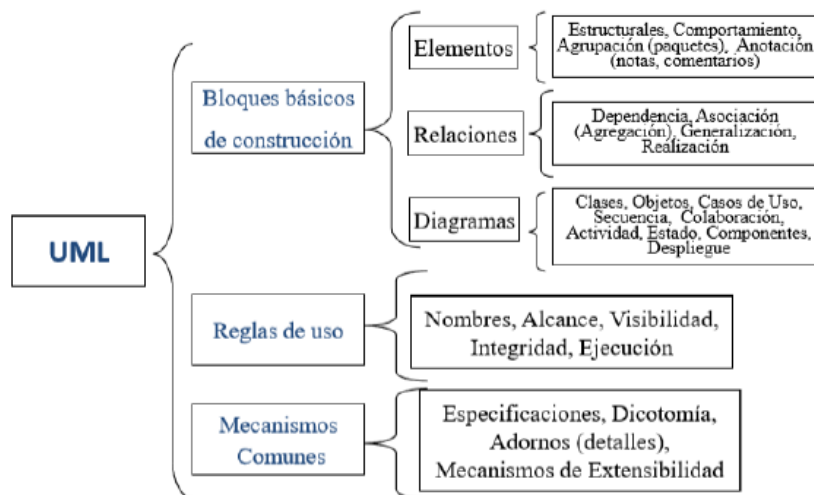
24) PASOS A CONSIDERAR EN EL PROCESO DE GESTIÓN DE RIESGOS Y BREVE DESCRIPCIÓN

Proceso de gestión del riesgo

- 1- **Identificar el riesgo:**
 - a. Identificar posibles riesgos de proyecto, negocio y producto.
 - b. Determinar en cada situación posibles riesgos.
- 2- **Estimación y evaluación de riesgo**
 - a. Determina la probabilidad y consecuencia del riesgo.
 - b. Probabilidad expresada en cuantitativa o cualitativa.
 - c. Consecuencias del tipo:
 - i. Seria.
 - ii. Catastrófica
 - iii. Tolerable
 - iv. Insignificante.
- 3- **Planificación del riesgo**
 - a. Trazar plan para evitar o minimizar ocurrencia de un riesgo.
 - b. Diseñan estrategias para evitar riesgos
 - c. Estas tareas se engloban y detallan en Plan de Gestión y Supervisión del riesgo.
- 4- **Control del riesgo**
 - a. Controla la ocurrencia de riesgos a lo largo del proyecto.
 - b. Asegura el cumplimiento de las tareas para evitar el riesgo.
 - c. Revisar periódicamente los riesgos identificados y sus consecuencias.
 - d. Los riesgos considerados deben ser discutidos en reuniones periodoncias



25) BLOQUES BASICOS DE CONSTRUCCIONES DE UML



Hay 4 tipos de **ELEMENTOS** en UML

- 1) Elementos Estructurales
- 2) Elementos de comportamiento
- 3) Elementos de agrupación
- 4) Elementos de anotación

1. Elementos Estructurales de UML (PREGUNTA DE EXAMEN)

| NOMBRE | IMAGEN | DESCRIPCIÓN |
|---|--------|---|
| Clase | | Conjunto de objetos que comparten los mismos atributos y operaciones. |
| Caso de USO | | <ul style="list-style-type: none"> - Visión de interacción que tiene una persona para usar el sistema. - Con línea discontinua no es CASO DE USO es colaboración. - Con línea discontinua y un reloj significa temporal, ejemplo: UN disparador cron que haga un Backup sin interacción del usuario. |
| INTERFAZ | | - Colección de operaciones que especifica un servicio de una clase o componente. |
| COMPONENTE | | Es un empaquetamiento físico de diferentes elementos como clases e interfaces. El componente es modificable mientras no sea un artefacto. |
| ARTEFACTO (PASAMOS DE ESTE SEGÚN LA TEACHER) | | Igual que el componente, pero ya no modificable. |
| NODO | | Artefacto principal que dan una visión de arquitectura física de sistema desde el punto de vista hardware y que protocolos de comunicación existen entre ellos. |

2. Elementos de Comportamiento de UML (PREGUNTA DE EXAMEN)

| NOMBRE | IMAGEN | DESCRIPCIÓN |
|---------------------------|--------|---|
| Interacción | | Conjunto de mensajes que se intercambian entre objetos. |
| Máquina de estados | | Especifica secuencia de estados por la que pasa un objeto en respuesta a un evento. |
| Estado | | - Colección de operaciones que especifica un servicio de una clase o componente. |
| Actividad | | Comportamiento que especifica la secuencia de pasos que ejecuta un proceso computacional. |

3. Elementos de agrupación de UML (PREGUNTA DE EXAMEN)

1. Son partes organizativas de los modelos UML
2. Hay un elemento de agrupación por principal llamado **paquete**, que es un artefacto de agrupación.
3. Los paquetes son puramente conceptuales (Solo existentes en tiempo de desarrollo)
4. Dentro de estos elementos podemos encontrar los **Frameworks**.

4. Elementos de Notación de UML (PREGUNTA DE EXAMEN)

1. Son las partes explicativas de los modelos UML
2. Hay un tipo principal llamado **nota** que son comentarios que se pueden aplicar para describir o hacer observaciones sobre un elemento del modelo.

UML PERMITE La visualización, especificación, construcción y documentación de un sistema a través de diferentes perspectivas. **!!!!ULTRA IMPORTANTE!!!!**

Hay 4 tipos de **RELACIONES** en UML

Estas nos permiten modelar el enlace entre diferentes modelos estructurales, mostrando información adicional como multiplicidad y nombres de roles, UML tiene estos tipos:

1. **Relación de Dependencia:** Una dependencia es una relación semántica entre dos clases donde si una clase cambia la otra clase puede ser afectada, se representa:

----->

3. **Relación de Generalización:** Es una relación de especialización donde los objetos de un elemento (HIJOS) son consistentes con los objetos de un elemento superior (PADRE), se representa:

HIJO ————▷ PADRE

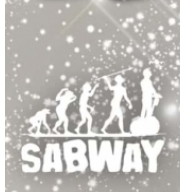
2. **Relación de Asociación:** Una asociación nos dice que conexiones hay entre los objetos. La **Agregación** es una clase especial que representa una relación de estructura entre un conjunto partes, se representa:

4. **Relación de Realización:** Es una relación semántica entre clasificadores (Clases) donde un clasificador especifica un contrato donde otro especificador garantiza llevar a cabo, se representa:

-----▷



15%
DE DESCUENTO
WWW.SABWAY.ES



15%
DE DESCUENTO
WWW.SABWAY.ES



15%
DE DESCUENTO
WWW.SABWAY.ES

26) TIPOS DE REQUISITOS Y SU DESCRIPCIÓN

- **Requisitos de Usuario:** Descripción de alto nivel, en lenguaje natural o diagramas, de lo que se espera que el sistema proporcione y las restricciones con las que debe funcionar. Sin comprometerse a como se debe hacer. Se describen usando lenguaje natural, tablas y diagramas, su principal problema es la falta de claridad.
- **Requisitos del Sistema:** Descripción detallada del sistema.
 - ✓ Deben ser precisos
 - ✓ Definir que se va a implementar.
 - ✓ Pueden ser parte del contrato entre el comprador y desarrollador.
- **Requisitos Funcionales:**
 - Son declaraciones de que debe proporcionar el sistema.
 - Especificar manera a reaccionar a determinada entrada.
 - Especificar como debe comportarse el sistema en diferentes situaciones.
- **Requisitos NO funcionales:** Valoraciones de circunstancias que afectan a todo el proyecto y se usan de manera global a todo el proyecto.
 - Son restricciones de los servicios que ofrece el sistema
 - Se aplican al sistema en su totalidad
 - Surgen de las necesidades el usuario
 - En general más críticos que los requisitos funcionales
 - Requisitos del producto:
 - Especifican comportamiento del producto.
 - Requisitos Organizacionales:
 - Provenir de 2 fuentes:
 - Organización propia del desarrollo.
 - Organización que ha requerido el desarrollo.
 - Requisitos Externos:
 - Se derivan de factores externos al sistema.
- **De Dominio**
 - ✓ Proviene del dominio y no de las necesidades del usuario.
 - ✓ Reflejan restricciones del dominio de la aplicación.

27) ESTRATEGIAS PARA APLICAR LAS PRUEBAS DEL SISTEMA *

1. Se comienza en la prueba de cada módulo, que la realiza el personal de desarrollo. **(PRUEBA UNIDAD)**
2. Con el esquema del diseño del software, los módulos probados se integran para comprobar sus interfaces en el trabajo conjunto. **(PRUEBA DE INTEGRACION)**
3. El software ya validado se integra con el resto del sistema (por ejemplo, elementos mecánicos, interfaces electrónicas, etc.) para probar su funcionamiento conjunto **(PRUEBA DEL SISTEMA)**
4. Por último, el producto final se pasa a la prueba de aceptación para que el usuario compruebe en su propio entorno de explotación si lo acepta como está o no **(PRUEBA DE ACEPTACIÓN)**

(PRUEBA UNIDAD)

La prueba de unidad puede abarcar desde un módulo hasta un grupo de módulos (incluso un programa completo).

Se trata de las pruebas formales que permiten declarar que un módulo está listo y terminado (no las informales que se realizan mientras se desarrollan los módulos).

(PRUEBA DE INTEGRACION) **

Están totalmente ligadas a la forma prevista de integrar los distintos componentes del software hasta contar con el producto global que debe entregarse e implican una progresión ordenada de pruebas que parte desde los componentes (módulos) y culmina en el sistema completo.

- Su objetivo es la prueba de las interfaces (flujos de datos) entre componentes o módulos.

Tipos fundamentales de integración:

- **Integración incremental:** Se combina el siguiente módulo que se debe probar con el conjunto de módulos que ya han sido probados. Se incrementa progresivamente el número de módulos hasta formar el programa completo. Tipos:

o Ascendente: Comienzan por los nodos hoja.

o Descendente: Comienzan por el nodo raíz.

- **Integración no incremental:** Se prueba cada módulo por separado y luego se integran todos de una vez y se prueba el programa completo. Se denomina también Big-Bang por que el número de módulos crece instantáneamente en la construcción del programa.

(PRUEBA DEL SISTEMA)

Es el proceso de prueba de un sistema integrado de hardware y software para comprobar lo siguiente:

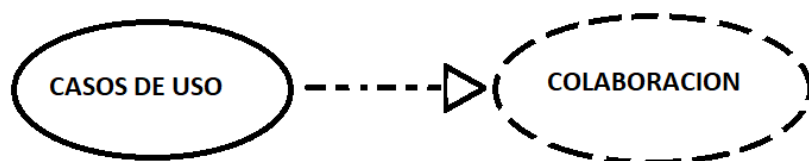
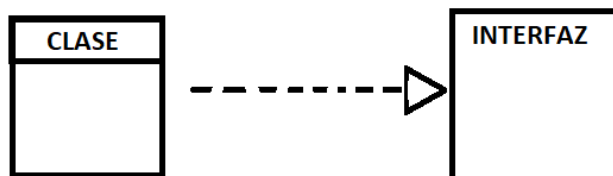
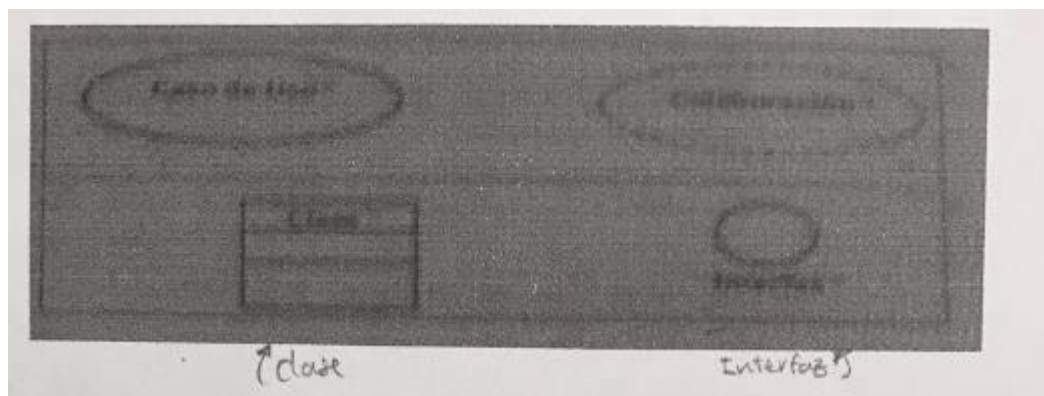
1. Cumplimiento de todos los requisitos funcionales, considerando el producto software final al completo en un entorno de sistema
2. El funcionamiento y rendimiento en las interfaces hardware, software, de usuario y de operador
3. Adecuación de la documentación de usuario
4. Ejecución y rendimiento en condiciones límite y de sobrecarga

(PRUEBA DE ACEPTACIÓN)

Es la prueba planificada y organizada formalmente para determinar si se cumplen los requisitos de aceptación marcados por el cliente. Sus características principales son las siguientes:

1. Participación del usuario
2. Está enfocada hacia la prueba de los requisitos de usuario especificados
3. Está considerada como la fase final del proceso para crear una confianza en que el producto es el apropiado para su uso en explotación

28) REPRESENTAR DE MANERA GRAFICA EL TIPO DE RELACIÓN QUE PUEDE EXISTIR ENTRE LAS DOS PAREJAS DE ELEMENTOS QUE APARECEN EN LA FIGURA E INTERPRETAR EL SIGNIFICADO DE DICHA RELACION EN EL MODELO CONCEPTUAL DE UML



RELACION: Realización: Es una relación semántica entre los clasificadores, en la cual un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. Es una mezcla entre dependencia y generalización.

RELACION: Realización: El objetivo de la arquitectura de un sistema es encontrar el conjunto mínimo de colaboraciones bien estructuradas que satisfacen el flujo de eventos, especificado en todos los casos de uso del sistema.