

PRACTICA Nº 1:

Se pide en el enunciado los siguientes rubros que hay que resolver.

Se agregan las reglas simples de UML 2.0 (que también existían en UML 1.0) que se explicaron en clase y se usaron en Programación II.

Diagrama de casos de uso

- Dibujar claramente los casos de uso, indicar si es un extend o un include. Todos los casos de uso tanto sea normal como alternativo o subcaso de uso tienen nombre que representan su objetivo.
- indicar claramente los actores del problema NO DE LA IMPLEMENTACION.

Caso de uso:

- Indicar que secciones (nombre de cada una) tienen sincronización, concurrencia y paralelismo y poner ese nombre dónde se describen.
 - debe estar claro en cada sección como se logra la sincronización y concurrencia.
- ciclos y alternativas
 - Siempre que hay una alternativa o la terminación de una condición de ciclo, debe quedar claro como se puede obtener el valor que se consulta. (ejemplo, si es el último en terminar de alguna manera tiene que haber acciones previas que permiten obtener esa información)
- acciones: llevan sujeto (quien realiza la acción), verbo (la acción) y predicado (sobre que o quien).
- la descripción se numera.

Diagrama de clases preliminar:

- SOLAMENTE contiene entidades o sea sustantivos del caso de uso. NO PUEDEN SER DE CONTROL EN EL DIAGRAMA DE COMUNICACIONES.
 - contiene las relaciones de composición, agregación y generalización
 - contiene la cardinalidad de la relación
 - NO CONTIENE RELACIONES QUE SOLO OCURREN EN EJECUCIÓN.

Diagrama de comunicaciones.

- SOLAMENTE CONTIENE OBJETOS DEL PROBLEMA **NO DE LA SOLUCIÓN.**
- Se muestran las interacciones con el/los actores (no importa que se implementen luego en la simulación como procesos) POR DEFINICIÓN EL ACTOR ES EXTERIOR AL SISTEMA, O SEA ES EL GENERADOR DE EVENTOS.
- Las únicas entidades del diagrama corresponden al diagrama de clases preliminar NO SE PUEDEN AGREGAR ENTIDADES, puede haber entidades del diagrama preliminar que no se requieran, ya que no tienen variables de instancia y/o métodos propios.
- Todos los métodos que se invocan
 - llevan el **nombre de la acción** del caso de uso y su **número en el caso de uso (puede tener múltiples)** (NO PUEDE INVENTAR METODOS SI NO ESTAN EN EL CASO DE USO)

- deben estar todas las acciones del caso de uso.
- SE DEBEN MANTENER LAS MISMAS RELACIONES ENTRE LAS ENTIDADES QUE EN EL DIAGRAMA DE CLASES PRELIMINAR.

Diagrama de clases final.

- Contiene todas las clases necesarias para instanciar a todos los objetos del diagrama de comunicaciones. Por definición: si la clase instancia un solo objeto (singleton) el objeto y la clase se llaman igual. Si instancia mas de uno es el nombre de la clase seguido de una palabra que indique su uso.
- Cada clase contiene
 - Su nombre
 - el tipo de objetos que instancia (entidad, control, interfaz, borde)
 - todos los métodos con el mismo nombre que en el diagrama de comunicaciones. (NI UNO MAS NI UNO MENOS).
 - cardinalidad de la relación.

Diagramas de secuencia.

- **Hay un diagrama por cada escenario posible** (en concurrencia y distribuidos que no son problemas secuenciales siempre hay diferentes situaciones para mostrar la concurrencia, sincronización y el paralelismo), y con los alternativos.
- El diagrama contiene todos los ciclos y alternativas del caso de uso:
 - todos los valores que determinan fin de ciclo o la decisión de la alternativa deben obtenerse claramente por medio de métodos o de la acción interna del método que ejecuta el ciclo/alternativa.
 - debe quedar claro si hay sincronización, exclusión mutua o concurrencia.
- El diagrama contiene LOS MISMOS METODOS (CON EL MISMO NOMBRE Y N° DEL CASO DE USO) del diagrama de comunicaciones y del diagrama de clases final.
- Los objetos tienen el mismo nombre que en el diagrama de comunicaciones y el mismo ícono, para asociarlo a su tipo. No pueden faltar ni puede haber nuevos objetos.
- Representa un diagrama de ejecución en el tiempo.

Objetos activos y pasivos:

- Marque los objetos activos en el diagrama de secuencia. Para cada objeto activo:
 - Indique el nombre de la interfaz para ese objeto activo.
 - Indique los métodos de la interfaz, que deben ser métodos del problema o sea del diagrama de comunicaciones / secuencia
 - Para cada objeto pasivo que usa la interfaz
 - Indique cual es el objeto pasivo
 - Indique cuales son los métodos que requiere de la interfaz.
 - Indique el nombre del componente que resuelve la interfaz.
 - Dibuje el conjunto. interfaz-componente.

Diagrama de secuencia de la solución.

- Muestre cual es el objeto pasivo y el método que invoca en la interfaz y agregue los objetos IPC necesarios para resolver el problema.
- Idem diagrama de secuencia normal con todos sus escenarios o sea sus diferentes situaciones.

Para el trabajo práctico

Para cada ejercicio de la práctica nº 1:

Se presentaran distintos requerimientos no funcionales del problema para determinar como varía su solución.

Se pide:

Para la versión 0

Lista de requerimientos NO FUNCIONALES del problema.

- lista de requerimientos no funcionales del enunciado en relación con los que se agreguen y justificación

Trabajando sobre una copia de los casos de uso y diagramas de la práctica Nº 1: Modificarlo para cumplir con los requerimientos no funcionales.

Caso de uso

Diagrama de clases preliminar

Diagrama de comunicaciones

Diagrama de clases final

Diagrama de secuencia

Objetos activos y pasivos

Diagrama de secuencia de la solución

Y se agregan:

Diagrama de clases de la solución.

- **Tiene las mismas reglas de un diagrama de clases.**
- Los IPC's se consideran clases del sistema operativo y llevan el nombre del IPC
- Los subsistemas se muestran como clases
- Se muestran las clases de los objetos activos, interfaz y componentes con sus respectivos métodos

Diagrama de subsistemas.

- Se marcan sobre el diagrama de secuencia de la solución los distintos subsistemas.
- Es un diagrama idéntico al de comunicaciones en los cuales los subsistemas incluyen los objetos pasivos internos e interactúan entre subsistemas por medio de las interfaces y componentes.
- Mantienen los mismo nombres definidos anteriormente.

Diagrama de implementación o de deployment.

- Se define sobre una red de computadoras, en cual computadora corre cada subsistema y se hace el diagrama correspondiente y sus interacciones.

Para la versión 1

Diagrama de secuencia de la solución distribuyendo subsistemas asincrónicos

- Se muestra en el diagrama la interfaz que pone en/saca de la cola y se agregan los objetos componentes que implementan los patrones de comunicaciones asincrónicos bidireccional en un ambiente cliente/servidor.

- Se agregan los componentes que implementan los patrones de comunicaciones de negociación y service discovery.

Diagrama de clases de la solución ampliado.

- **Tiene las mismas reglas de un diagrama de clases.**
- Los IPC's y las comunicaciones se consideran clases del sistema operativo y llevan el nombre del IPC/comunicaciones TCP/IP.
- Los subsistemas se muestran como clases
- Se muestran las clases de los objetos activos, interfaz y componentes con sus respectivos métodos

Diagrama de subsistemas ampliado.

- Se marcan sobre el diagrama de secuencia de la solución los distintos subsistemas.
- Es un diagrama idéntico al de comunicaciones en los cuales los subsistemas incluyen los objetos pasivos internos e interactúan entre subsistemas por medio de las interfaces y componentes.
- Mantienen los mismo nombres definidos anteriormente.

Diagrama de implementación o de deployment corregido.

- Se define sobre una red de computadoras, en cual computadora corre cada subsistema y se hace el diagrama correspondiente y sus interacciones.

Para la versión 2

Diagrama de secuencia de la solución distribuyendo subsistemas sincrónicos

- Se muestra en el diagrama la implementación del algoritmo que permite la distribución de los elementos de los componentes sincrónicos. Se implementan los patrones de comunicaciones con un broker (forwarding), de suscripción/notificación y de registración de servicio. Se adecuan los servicios de los patrones desarrollados en la versión anterior.

Diagrama de clases de la solución ampliado.

- **Tiene las mismas reglas de un diagrama de clases.**
- Los IPC's se consideran clases del sistema operativo y llevan el nombre del IPC y sistemas de comunicaciones.
- Los subsistemas se muestran como clases
- Se muestran las clases de los objetos activos, interfaz y componentes con sus respectivos métodos

Diagrama de subsistemas ampliado.

- Se marcan sobre el diagrama de secuencia de la solución los distintos subsistemas.
- Es un diagrama idéntico al de comunicaciones en los cuales los subsistemas incluyen los objetos pasivos internos e interactúan entre subsistemas por medio de las interfaces y componentes.
- Mantienen los mismo nombres definidos anteriormente.

Diagrama de implementación o de deployment corregido.

- Se define sobre una red de computadoras, en cual computadora corre cada subsistema y se hace el diagrama correspondiente y sus interacciones.

Para la versión 3

Diagrama de secuencia de la solución distribuyendo subsistemas sobre un MOM orientado a servicios

- Se implementan los patrones de comunicaciones con un broker (forwarding), de suscripción/notificación y de registración de servicio. Se adecuan los servicios de los patrones desarrollados en la versión anterior.

Diagrama de clases de la solución ampliado.

- **Tiene las mismas reglas de un diagrama de clases.**
- Los IPC's se consideran clases del sistema operativo y llevan el nombre del IPC y sistemas de comunicaciones.
- Los subsistemas se muestran como clases
- Se muestran las clases de los objetos activos, interfaz y componentes con sus respectivos métodos

Diagrama de subsistemas ampliado.

- Se marcan sobre el diagrama de secuencia de la solución los distintos subsistemas.
- Es un diagrama idéntico al de comunicaciones en los cuales los subsistemas incluyen los objetos pasivos internos e interactúan entre subsistemas por medio de las interfaces y componentes.
- Mantienen los mismo nombres definidos anteriormente.

Diagrama de implementación o de deployment corregido.

- Se define sobre una red de computadoras, en cual computadora corre cada subsistema y se hace el diagrama correspondiente y sus interacciones.