

Instrucciones.

Realizar el análisis usando UML. El análisis debe incluir:

- Diagrama de Clases y diagramas de instancias en aquellos casos en que sea conveniente.
- Diagramas de interacción . Diagramas de estados para las clases que lo precisen.
- Diagramas de casos de uso y su especificación.
- Otros elementos de modelado que sean oportunos en el problema en cuestion.

Problemas.

1.- Se desea automatizar un aparcamiento, con capacidad para 327 automóviles, de acuerdo a los siguientes requisitos:

Los usuarios del aparcamiento dispondrán de una tarjeta magnética donde figura registrado su código de identificación. A su llegada al aparcamiento, el usuario introducirá la tarjeta en el lector correspondiente, lo que hace que se eleve la barrera situada en la entrada. Esta barrera permanece levantada un cierto tiempo, descendiendo luego automáticamente. Para salir del aparcamiento se procede de igual forma con la barrera situada a la salida.

Tanto las entradas como las salidas deben quedar registradas con objeto de realizar periódicamente una facturación a los usuarios, según el tiempo de aparcamiento consumido. Estas facturas se emitirán a petición del operador.

El sistema debe tener en cuenta la ocupación del aparcamiento, controlando un semáforo situado a la entrada. Si hay plazas libres, el semáforo debe estar verde, pasando a rojo si el aparcamiento se llena. Además, cuando el aparcamiento esté lleno no debe permitirse la entrada de nuevos vehículos.

2.- Se desea informatizar el Departamento Editorial de una determinada revista. A continuación se especifica cuál es el funcionamiento que se pretende establecer.

Los autores (o sus representantes) deberán entregar los originales en soporte magnético, ya sea por correo electrónico, ordinario o en persona, en la Editorial de la revista, donde serán revisados para su aceptación o rechazo. La tasa de rechazos es alta, e incluso muchos de los artículos que finalmente se aceptan deben sufrir alguna modificación antes de que la versión definitiva se envíe al Departamento de Maquetación.

Todos los artículos que llegan a la Editorial se guardan en una base de datos. Además, los artículos recibidos deben ser registrados, identificándolos con un número de referencia que servirá para su localización posterior. Este número de referencia se comunica al autor mediante una confirmación de recepción enviada por los mismos medios por los que se recibió el original. A continuación, cada artículo es enviado por correo electrónico a tres revisores externos para su evaluación. Cada revisor aconseja la aceptación o el rechazo del artículo o bien sugiere una serie de modificaciones que deberán ser realizadas por el autor. Un artículo se acepta si al menos dos revisores lo aceptan tal cual está, mientras que se rechaza si al menos dos revisores lo rechazan. Tanto las aceptaciones como los rechazos se comunican por escrito al autor. Los artículos aceptados se envían a Maquetación para que se preparen con ellos los próximos números de la revista.

Si el resultado de la evaluación no determina la aceptación o el rechazo del artículo, se envían al autor las sugerencias de los revisores, tras lo cual el autor puede modificar el artículo y volver a presentarlo a la Editorial. En este caso, se mantiene el número de referencia del artículo y se envía para su evaluación a los mismos revisores que evaluaron la primera versión.

Además del Registro de Entradas de artículos, el departamento editorial mantiene un archivo de autores. Cuando un artículo llega por primera vez a la editorial, se actualizan los datos del autor en el archivo o bien se crea un nuevo registro en el archivo, si se trata de un autor novel. La información del archivo de autores se utiliza para notificar a éstos el resultado de la revisión.

El sistema debe registrar el estado de cada artículo, de forma que esta información pueda ser consultada por el personal de la Editorial, por ejemplo a petición del autor.

3.- Un sistema central de reservas de alojamiento recibe peticiones de reserva (zona, categoría del hotel, número y tipo de habitaciones a reservar, régimen de alojamiento, fechas de entrada y salida y los datos del cliente. Con esta información busca un hotel que disponga de las habitaciones que pide el cliente y

desencadena una petición al hotel para ver si dispone de las plazas libres requeridas. Este proceso se repite hasta que se encuentre un hotel de las características deseadas con plazas libres o hasta que no haya más hoteles. Si se encuentra hotel, se formaliza la reserva y se imprime un documento de reserva de habitaciones para el cliente.

4.- Se pretende desarrollar el software de un **sistema telefónico** convencional, de acuerdo a la siguiente especificación informal de requisitos:

a) Los teléfonos son manejados por una serie de usuarios que pueden descolgarlos, colgarlos o marcar un número a voluntad. El sistema debe ser fiable y no exigir un comportamiento predeterminado por parte de los usuarios.

Para simplificar se supondrá que el sistema carece de centralitas, estando cada teléfono conectado con todos los demás mediante una línea directa.

Cuando un usuario desea usar un teléfono A para realizar una llamada, lo primero que debe hacer es descolgarlo. Entonces recibirá por el auricular el tono de línea. A continuación deberá marcar el número del teléfono B al que quiere llamar. Si pasado un cierto tiempo no se ha recibido respuesta del teléfono B, esto puede deberse a que esté averiado o a que no exista. En ambos casos se generará un tono de error en A hasta que se cuelgue. Por el contrario, si el teléfono B está ocupado, rechazará la llamada, generándose entonces el tono de ocupado en A hasta que se cuelgue de nuevo. Por último, si B está libre, aceptará la llamada, generándose un tono de llamada en A mientras suena el timbre en B.

Si ningún usuario responde a la llamada en B, la conexión finalizará cuando A cuelgue, desapareciendo entonces el tono de llamada en A y el timbre en B.

Cuando se descuelgue el teléfono B se establece la comunicación entre ambos usuarios. Si B cuelga su teléfono la comunicación no debe cortarse, sino que se interrumpe temporalmente hasta que B descuelgue de nuevo.

Para cortar la comunicación es necesario que se cuelgue el teléfono que realizó la llamada (en este caso el A). En ese momento el teléfono B genera tono de ocupado hasta que se cuelgue.

b) Se desea ampliar las prestaciones del sistema anterior ofreciendo una función de Llamada en espera. Para ello, se dota a los teléfonos de una señal luminosa que permite avisar al usuario de

la existencia de una nueva llamada, incluso cuando esté hablando, y de un botón que permite aceptar llamadas en espera.

Supongamos que los teléfonos A y B están estableciendo una comunicación entre sí, sin importar quién haya iniciado la llamada. Si desde un teléfono C se intenta comunicar con uno de ellos (por ejemplo el A), se generará en C el tono de ocupado, como se indicó anteriormente. Sin embargo, en el teléfono A deberá encenderse el indicador de llamada en espera mientras el usuario en C no cuelgue su teléfono.

Para aceptar una llamada en espera, el usuario de A debe pulsar la tecla correspondiente. En ese momento se establece comunicación entre A y C, mientras que la comunicación con B queda temporalmente suspendida, generando B tono de ocupado. Para finalizar una llamada de este tipo basta con que A o C cuelguen su teléfono, restableciéndose entonces la comunicación con B (salvo en el caso en que éste hubiese colgado el teléfono con anterioridad).

Adecuar el análisis a la nueva situación, intentando que la especificación de la nueva función se traduzca en una ampliación o extensión de lo anteriormente hecho y no en una modificación substancial.

5.- Se desea desarrollar un sistema autónomo de control del nivel de azúcar en sangre para pacientes diabéticos, según la especificación siguiente:

El sistema cuenta con un sensor que le informa periódicamente del nivel de azúcar en sangre. Además, controla un dispositivo que permite inyectar varios productos al paciente, con el fin de regular su nivel de azúcar. Cada vez que recibe información del nivel de azúcar del paciente, el sistema debe calcular cuál es el producto a inyectar y la dosis adecuada, así como un tiempo de espera hasta que se produzca la reacción. De esta forma se evita inyectar repetidamente el producto, antes de que el paciente reaccione al mismo. Para calcular la dosis y el tiempo de espera el sistema debe conocer las dosis inyectadas con anterioridad, de forma que prevea la existencia de dosis residuales en el paciente. El sistema debe monitorizar los niveles de azúcar incluso cuando está en espera, por si superan unos márgenes de seguridad establecidos, en cuyo caso debe actuar nuevamente.

6.- Una determinada compañía aérea desea desarrollar un sistema de reserva y venta de billetes por ordenador (CRS) de forma que se agilicen estas operaciones. A partir de las conversaciones mantenidas con su Director de Operaciones, se ha elaborado la siguiente lista de requisitos:

- ? *La compañía realiza vuelos a diferentes destinos. La información referente a cada vuelo (origen, destino, fechas, horas, tarifa y número de plazas) se introduce en el sistema mediante una aplicación de carga de datos ya existente y que no es necesario considerar aquí.*
- ? *El sistema debe encargarse de las reservas, ventas y devoluciones de billetes y de las listas de espera. Estas operaciones se realizan a través de agencias de viajes y oficinas comerciales de la compañía en los aeropuertos, todas ellas conectadas vía módem al CRS.*
- ? *Las reservas no llevan cargo alguno, pero deben transformarse en ventas un mínimo de 24 horas antes de la salida del vuelo. En caso contrario son anuladas automáticamente.*
- ? *La información de ventas de billetes se envía al sistema de Contabilidad de la compañía, también informatizado (no es necesario modelarlo en este problema).*
- ? *Cualquier billete vendido puede ser anulado hasta 24 horas **después** de la salida del vuelo, debiendo informarse de esto a Contabilidad para que se realice la devolución del importe.*
- ? *Si un vuelo está completo, ya sea con reservas o ventas, un cliente puede solicitar ser incluido en lista de espera. La anulación de una reserva o venta en un vuelo completo hace que el primer cliente de la lista de espera pase a la situación de reserva, siguiendo a partir de ese momento las condiciones generales de la reserva.*
- ? *A través de los terminales de facturación del aeropuerto se controlará la ocupación real del vuelo. Los billetes no facturados 20 minutos antes de la salida, son anulados por la compañía, vendiéndose las plazas correspondientes al primer cliente que las solicite.*
- ? *Los billetes vendidos y **no utilizados** pueden ser anulados hasta 24 horas después de la salida del vuelo. En este caso, se informará a Contabilidad para que realice la devolución del importe.*

7.- *Supongamos una estructura de datos a la que vamos a denominar PIPE, que nos va a permitir la transferencia de datos entre procesos en orden FIFO, de forma que nos permita establecer la comunicación entre procesos sin que estos sepan qué procesos están en el otro extremo de la pipe. Los datos son leídos en el orden en que fueron escritos en la pipe, y no se permiten cambios en ese orden.*

Una pipe ha de ser creada antes de poder ser utilizada, pero una vez que ésta existe, pueden utilizarla tantos procesos como se quiera. Una vez que una pipe está creada, si un nuevo proceso

quiere usarla, no tiene más que “abrirla”, bien para lectura o para escritura. Podemos ver la pipe como una estructura en la que unos procesos escriben en un extremo y otros van leyendo del otro extremo. Una vez que un proceso deja de necesitar la pipe debe cerrarla.

Apertura de pipes

Si un proceso abre una pipe para lectura y hay algún proceso escritor, entonces la operación se completa; en caso contrario, el proceso deberá esperar (dormir) hasta que un proceso abra la pipe para escritura. El mismo caso se presenta cuando un proceso abre una pipe para escritura; la operación se realizará si hay procesos lectores, o el proceso dormirá hasta que los haya. No tiene sentido para una pipe ser abierta para lectura si no hay posibilidad de que contenga datos, y lo mismo para escritura.

Lectura y escritura de pipes

El tamaño de la pipe es limitado. En caso de que un proceso intente escribir en la pipe un dato, se realiza la escritura si hay espacio. Si un proceso intenta escribir en una pipe en la que no hay espacio, éste se va a dormir, esperando que algún proceso deje espacio libre en la pipe con alguna lectura. Si un proceso intenta leer en una pipe vacía, el proceso lector dormirá hasta que otro proceso escriba datos en la pipe. Si la pipe no está vacía, se realiza la lectura.

Cierre de pipes

La destrucción de la pipe se realizará de forma controlada. Cuando no queden procesos que tengan abierta la pipe para escritura, se responderá normalmente a la operaciones de lectura mientras queden datos en la pipe; una vez que la pipe esté vacía se responderá con un mensaje de error. Por otra parte, cualquier intento de escritura en una pipe en la que no queden lectores devolverá un error. La pipe se destruirá cuando no haya lectores ni escritores.

8.- Realizar el análisis de requisitos de un sistema de control para los ascensores de la ETSI Informática, según la especificación preliminar que se describe a continuación:

El sistema de ascensores consta de dos ascensores que se encargan de transportar personas de una planta a otra en un edificio de cuatro plantas (planta baja y tres pisos). El sistema de control de los ascensores se encarga de planificar el movimiento de éstos para que se satisfagan de forma eficiente las peticiones de servicio de los usuarios.

En cada planta existen dos botones de petición de servicio: uno para subir y otro para bajar, (salvo la planta baja y la tercera, por sus características especiales). Estas peticiones podrían ser satisfechas indistintamente por cualquiera de los dos ascensores.

En el interior de cada ascensor hay un panel con cuatro botones de forma que cada usuario puede indicar cuál es su planta de destino. Al pulsar cualquiera de estos botones, se ilumina, de forma que los pasajeros pueden comprobar que su petición ha sido registrada. Al llegar a la planta en cuestión el indicador luminoso parpadeará mientras las puertas estén abiertas y se apagará cuando el ascensor abandone dicha planta.

Los ascensores tienen que funcionar de forma eficiente y razonable. Por ejemplo, si alguien llama al ascensor pulsando el botón de bajada en la segunda planta, el ascensor que pase a continuación por dicha planta y lleve sentido de bajada, debe parar para cargar al pasajero. Por otra parte, si un ascensor está desocupado (no tiene ninguna petición pendiente), debe permanecer parado en la planta donde se encuentre. Un ascensor no debe cambiar el sentido de su movimiento hasta que los pasajeros que quieran viajar en el sentido actual hayan llegado a su destino. Un ascensor lleno no debe aceptar más peticiones de servicio (para determinar si están llenos, los ascensores disponen de un sensor de peso).

Cada ascensor dispone de un sensor que le permite detectar cuando llega a cada una de las plantas.

El sistema de control de los ascensores los moverá de acuerdo con la planificación de las peticiones que realice enviando señales de subir, bajar y parar a los motores de cada ascensor.

9.- Queremos desarrollar una herramienta case basada en la metodología OMT. Entre otras cosas esta herramienta debe permitir la representación del comportamiento de los objetos mediante diagramas de estados (DEs). Una de las funciones de la herramienta será controlar la ejecución de un prototipo del sistema a partir de su especificación OMT, de modo que el usuario pueda depurar esta especificación. La herramienta dispondrá de un planificador que accederá al DE de cada uno de los objetos del sistema, determinando qué objetos tienen transiciones disparables. A continuación se seleccionará de forma aleatoria una de estas transiciones y se ejecutará, enviando los mensajes que se especifiquen en dicha transición y realizando las acciones y actividades asociadas al disparo de la transición.

10.- Se pretende desarrollar un programa que simule el funcionamiento de un banco, para obtener información para la organización del mismo. Se busca un compromiso entre la atención rápida al cliente y un número de empleados lo más reducido posible. El funcionamiento que debe simularse se describe a continuación:

Al principio de la simulación, el usuario indica el número de empleados que hay en el banco. Cuando un cliente entra al banco, lo que se produce en intervalos aleatorios, comprueba si hay algún empleado libre. Si es así, acude a él para ser atendido. En caso contrario, el cliente se pone a hacer cola hasta ser atendido. Cada empleado está ocupado atendiendo a un cliente durante un tiempo variable que oscila aleatoriamente alrededor de un valor promedio propio de ese empleado. Si, por el contrario, la cola está vacía, espera a ser solicitado por algún cliente que entre al banco.

El programa debe simular el funcionamiento del banco durante una jornada de trabajo, generando, al final, estadísticas del número total de clientes atendidos, el porcentaje de clientes que han tenido que hacer cola, el tiempo medio de espera en la cola, el número de clientes atendidos por cada empleado y el porcentaje de tiempo que ha estado ocupado cada empleado.

11.- *Un vehículo dispone de un sistema de control integral que realiza las funciones de cambio automático, control de tracción, control de frenada y velocidad de crucero.*

El sistema posee sensores de velocidad global del vehículo, velocidad de cada rueda, de posición de los pedales (freno y acelerador), de posición de la palanca de cambio, y además un mecanismo de control del acelerador.

? *El cambio automático funciona cuando la palanca está en la posición AUTO de este modo:*

La velocidad a la cual se cambia de una marcha a la siguiente no es fija sino que depende de la posición del acelerador. Cuanto más a fondo se encuentre pisado más tarde se producirá el cambio a la marcha siguiente (se apura la marcha).

Si se pisa a fondo el acelerador y después se suelta el sistema reduce una marcha.

Si se pisa el freno, las marchas se van reduciendo a medida que disminuye la velocidad.

? *El control de tracción suelta un poco el acelerador si la velocidad de las ruedas motrices es mayor que la de las otras ruedas estando el acelerador pisado.*

- ? *El control de frenada suelta durante un instante el freno de alguna rueda si se bloquea esta mientras las demás giran a una velocidad superior a una preestablecida estando el freno pisado.*
- ? *La función de velocidad de cruce se activa mediante un botón situado en el volante y activa un indicador en el cuadro. Esta función consiste en un mecanismo que acelera o frena el vehículo para que mantenga constante la velocidad que tenía cuando se pulsó el botón. Si se pulsa el acelerador cuando el sistema está activo el vehículo puede circular más rápido pero cuando se suelte el vehículo vuelve a la velocidad fijada. Esta función se desactiva cuando se pulsa de nuevo el botón o se pisa el freno.*

12.- Se pretende desarrollar un sistema software para simular el comportamiento de un ordenador, como paso previo a la fabricación del mismo. La arquitectura y funcionamiento del ordenador se describen a continuación.

El ordenador consta de:

- ? *memoria principal (que no aparece en la figura y no es necesario modelar), donde se almacenan instrucciones y datos, y que está controlada por una unidad de control de la memoria (UCM).*
- ? *unidad de control (UC), que contiene un contador de programa (CP). La UC ejecuta directamente las instrucciones WRITE, SAL, SCND y RETURN.*
- ? *una unidad aritmético-lógica (UAL), dotada de un registro acumulador (AC), que ejecuta las instrucciones LOAD, ADD, SUBS y RESET.*

Las instrucciones se componen de un código de operación y de una dirección de operando.

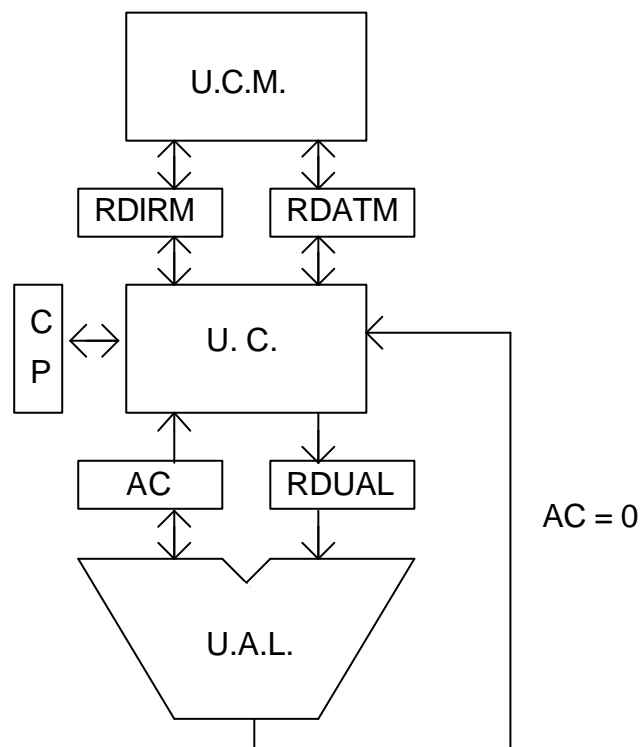
El lenguaje ensamblador que utiliza este sistema consta de las siguientes instrucciones:

- ? *LOAD M Cargar el contenido de la posición de memoria M en el acumulador.*
- ? *ADD M Sumar el contenido de la posición de memoria M al acumulador.*
- ? *SUBS M Restar el contenido de la posición de memoria M al acumulador.*
- ? *RESET Resetear el contenido del acumulador.*
- ? *WRITE M Escribir el contenido del acumulador en la posición de memoria M.*

- ? *SAL I Salto incondicional a la instrucción almacenada en la posición I.*
- ? *SCOND I Salto a la instrucción I si el contenido del acumulador es cero. Para detectar esta condición, existe una señal, mediante la cual la UAL informa del estado del acumulador tras la ejecución de una instrucción LOAD, ADD o SUBS y RESET.*
- ? *RETURN Fin de programa.*

Además de los elementos citados, el ordenador dispone de los siguientes registros para conectar sus componentes:

- ? *Registro de direcciones de memoria (RDIRM), mediante el cual la UC informa a la UCM de qué dirección de memoria quiere leer o escribir.*
- ? *Registro de datos (RDATM), mediante el cual la UC indica el valor a escribir o recibe el valor leído de la posición de memoria indicada por RDIRM.*



- ? *Registro de datos (RDUAL), mediante el que la UC indica a la UAL el valor del operando que debe usarse para la ejecución de una instrucción LOAD, ADD o SUBS.*

Las operaciones de lectura/escritura en memoria se realizan de la siguiente forma:

Lectura

La UC pone en RDIRM la dirección de memoria que quiere leer. La UCM detecta la existencia de un dato en RDIRM y realiza la lectura, que dura un tiempo indeterminado, tras el cual la UCM pone el contenido de esa dirección en RDATEM. La UC detecta la existencia de un dato en RDATEM y lo lee.

Escritura

La UC pone en RDIRM la dirección de memoria que quiere escribir y en RDATEM el valor que quiere almacenar en dicha dirección. La UCM detecta la existencia de datos en ambos registros y realiza la escritura, que dura también un tiempo indeterminado. Para informar a la UC de que la escritura ha finalizado y la memoria esta lista para una nueva operación, la UCM inicializa el registro RDIRM.

Ciclo de ejecución de una instrucción

Consta de los siguientes pasos:

- ? *Búsqueda de la instrucción indicada por el contador de programa CP.*
- ? *Descodificación de la instrucción.*
- ? *Búsqueda del operando (si es necesario) o escritura en memoria (para la instrucción WRITE).*
- ? *Ejecución de la instrucción en la UAL (para las instrucciones LOAD, ADD, SUBS y RESET).*
- ? *Incremento o modificación del contador de programa CP.*

13.- Se desea desarrollar un administrador de tareas para un sistema operativo multitarea, de acuerdo a la siguiente especificación preliminar:

El administrador de tareas recibe del usuario la orden de que debe ejecutar un determinado proceso. El administrador de tareas solicita al administrador de ficheros dicho proceso, lo carga en memoria y lo inicializa, dejándolo en estado Dormido. Los procesos que están dormidos pasan al estado Preparado cuando se encuentran disponibles los recursos necesarios para su ejecución. El administrador de tareas, elige para su ejecución un proceso de los que están en la cola de preparados y lo envía al procesador. Periódicamente se genera una interrupción que hace que el proceso en ejecución salga del procesador y vuelva a la cola de preparados. Si mientras está en ejecución, un proceso necesita algún recurso adicional, sale del procesador y pasa a cola de

dormidos, liberando todos los recursos que tuviese asignados (con lo que se evitan posibles bloqueos). El usuario puede realizar una petición del estado del sistema, cuya respuesta consistirá en una lista de los procesos que están dormidos, otra de procesos preparados y otra de los recursos que están disponibles. Además el usuario puede cancelar un determinado proceso o reinicializar todo el sistema (cancelar todos los procesos).

14.- Se quiere desarrollar un sistema de correo electrónico para el VAX situado en el Centro de Cálculo de la ETSI Informática. Este sistema de mensajería constará de dos elementos:

- ? *un Servidor Local de correo. Se trata de un proceso que se está ejecutando continuamente (salvo en caso de fallo imprevisto) en el VAX y que se comunica - enviando y recibiendo mensajes de correo - con el Servidor de correo de la Universidad, situado en el Servicio Central de Informática en el campus de Teatinos (considerar el Servidor de la Universidad como una entidad externa de nuestro sistema).*
- ? *una aplicación de gestión de correo, llamada EAN, que cada usuario puede ejecutar desde su cuenta del VAX tecleando el comando **ean**. Para salir de la aplicación se utiliza el comando **quit**.*

El Servidor Local se encarga de recibir mensajes provenientes del servidor de la universidad, comprobar que el destinatario de los mensajes tiene una cuenta en el VAX (en caso contrario los mensajes se devuelven, indicando que el destinatario es desconocido), remitir a los usuarios los mensajes recibidos, lo que se hará cuando los usuarios se conecten a la aplicación EAN (no puede hacerse en el momento en que el Servidor recibe el mensaje porque el Servidor no sabe cuál es el fichero de mensajes de cada usuario, con lo que no puede escribir directamente en él) y enviar al servidor de la universidad los mensajes escritos por los usuarios del VAX cuyos destinatarios no sean locales, es decir, no sean usuarios del propio VAX.

Por otra parte, cada usuario dispone de un fichero de mensajes, donde se almacenan tanto los mensajes que envía como los que recibe. Al entrar en la aplicación EAN, hay que informar al Servidor Local, para que éste nos envíe los mensajes recibidos desde la última conexión. Aparte de esto, EAN dispone de los siguientes comandos:

- ? ***list.** Lista las cabeceras (número de mensaje, destinatario, remitente y título) de los mensajes que contiene nuestro fichero de mensajes.*
- ? ***print(n),** donde n es un número de mensaje. Muestra por el terminal el contenido de un mensaje.*
- ? ***delete(n),** elimina un mensaje del fichero.*
- ? ***send.** Llama al editor de textos para escribir un mensaje. Al salir del editor con <ctrl Z>, el mensaje que hemos escrito se envía al Servidor Local y queda además almacenado en nuestro fichero. Si en un cierto tiempo no hemos recibido confirmación de que el Servidor ha recibido el mensaje que acabamos de enviar, hay que informar al usuario de que el servidor no está activo (y por tanto el mensaje no ha sido enviado), mediante el mensaje *MTA failure*.*
- ? ***accept.** Si, mientras estamos usando el EAN, el Servidor Local recibe algún mensaje para nosotros, debemos ser informados de ello inmediatamente, mediante el mensaje *Mail at tecma1.uma.es*. Entonces podemos usar el comando **accept** para que el mensaje recibido se grabe en nuestro fichero de mensajes. (Hay que tener en cuenta que el Servidor Local no sabe en qué terminal estamos conectados, por lo que no puede escribir directamente en él).*

15.- El modelo orientado a objetos parece, al menos inicialmente, adecuado para su implementación sobre sistemas concurrentes o distribuidos. Sin embargo, existen algunos problemas a la hora de diseñar un lenguaje orientado a objetos concurrente (LOOC), entre los que podemos citar la necesidad de establecer mecanismos de sincronización entre los objetos. Algunas de las propuestas de LOOC se basan en la extensión de un LOO secuencial mediante una serie de clases que posibiliten la concurrencia.

a) Especificar, mediante OMT, las clases necesarias para extender un LOO secuencial, teniendo en cuenta los siguientes requisitos:

- ? Los objetos que describe el lenguaje deben ser considerados como procesos. Para ello, todas las clases heredarán de una clase *Proceso*. Al crearse un objeto/proceso, éste ejecutará inicialmente las instrucciones de su método de creación, quedando a continuación inactivo hasta que reciba una petición (en forma de mensaje). Cuando un objeto recibe un mensaje, ejecutará el método correspondiente, volviendo a quedar inactivo cuando finalice dicho método. Cuando un objeto necesite sincronizarse con otro, deberá ejecutar una primitiva *wait* (que figurará como instrucción en el método que está

ejecutando). Si esta operación se resuelve con éxito, el objeto continuará ejecutándose. Si no, pasará a un estado de espera, del que será despertado cuando la continuación sea posible.

Nota: Supondremos que los métodos se ejecutan de forma atómica, es decir, dos mensajes consecutivos que se envían a un objeto se ejecutan secuencialmente, sin que exista entrelazado entre ellos. Esta restricción equivale a suponer que dentro de cada objeto existe un único *thread* o flujo de control y que cada objeto tiene asociada una cola de mensajes pendientes de resolver.

? El mecanismo de sincronización que vamos a utilizar consistirá en el uso de semáforos. Un semáforo consiste en una variable entera no negativa S , sobre la que pueden realizarse dos operaciones:

wait(S). Si $S > 0$, entonces decrementar S . Si no, suspender la ejecución del proceso que ha realizado la operación.

signal(S). Si hay procesos suspendidos en este semáforo, despertar al primero de ellos. Si no, incrementar el valor del semáforo.

Inicialmente, el semáforo toma un valor entero que se estipula en su método de creación.

b) Utilizar las clases definidas anteriormente para especificar, usando la técnica OMT, un sistema de productores/consumidores con buffer acotado:

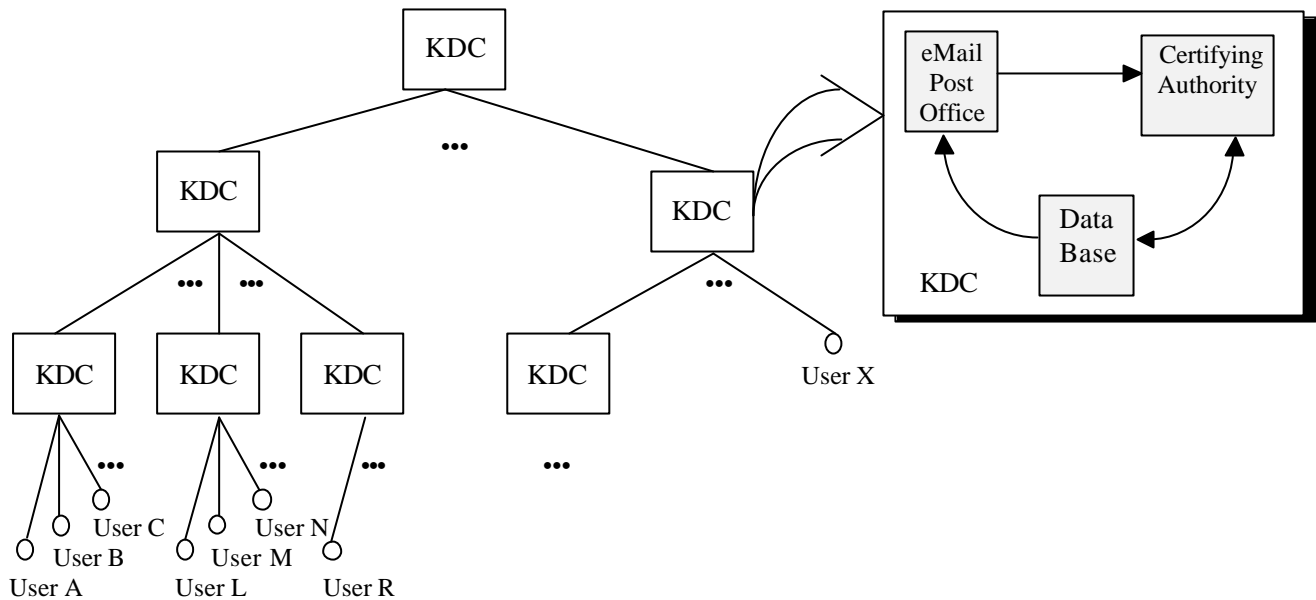
? Los productores generan ítems, que almacenan en un *Buffer* de tamaño limitado N . Si el *Buffer* está lleno, el productor deberá suspender hasta que haya espacio de nuevo.

? Los consumidores retiran ítems del *Buffer*. Si éste está vacío, el consumidor debe suspender hasta que existan de nuevo elementos en el *Buffer*.

Nota: Para resolver este problema, es necesario utilizar dos semáforos, uno de ellos, *Elementos*, indicando el número de elementos actualmente en el *Buffer*, para sincronizar los consumidores, y el otro, *Huecos*, indicando el número de posiciones libres, para sincronizar los productores.

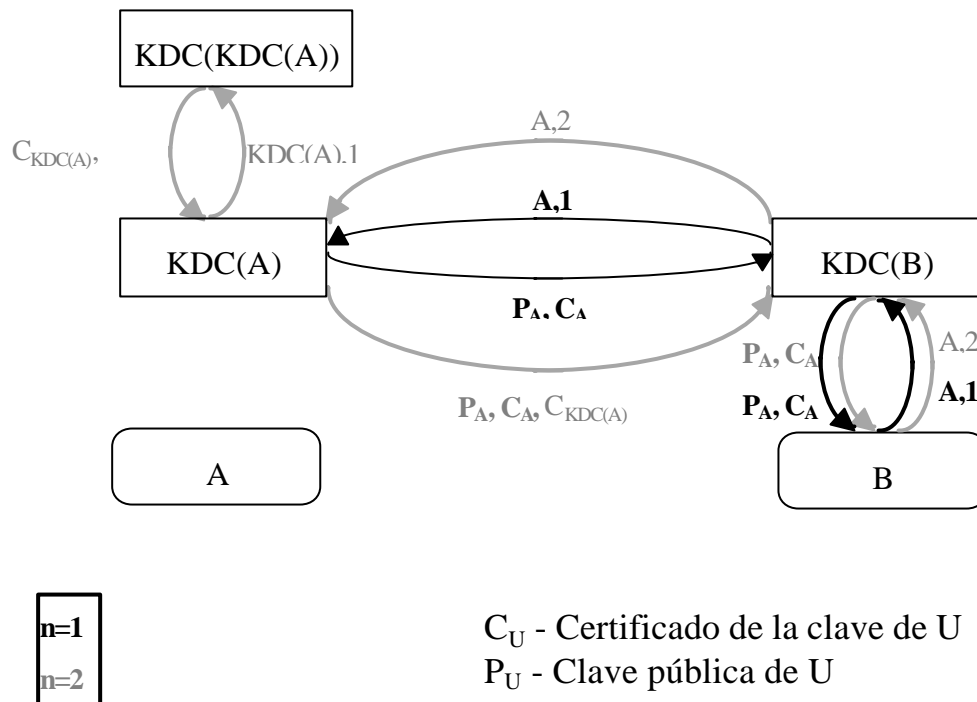
16.- Vamos a realizar un sistema jerárquico de distribución de claves para correo electrónico cifrado mediante un criptosistema de clave pública. En este tipo de sistemas cada usuario posee dos claves: una pública que se utiliza para cifrar mensajes que van dirigidos a ese usuario y otra privada que se utiliza para que el usuario pueda descifrar los mensajes que le llegan. Por tanto, para enviar un mensaje a un usuario es necesario conocer su clave pública.

En el esquema propuesto cada grupo de usuarios registra su clave pública en su servidor de claves (KDC) que forma parte de una estructura jerárquica:



Cada KDC es responsable del mantenimiento de las claves de sus usuarios y de certificar que dichas claves son válidas.

Cuando un usuario **B** desea obtener la clave de otro **A**, **B** debe solicitar esa clave a su KDC. Si el usuario **A** pertenece al KDC, éste envía directamente la clave a **B**. En caso contrario, el KDC redirige la petición directamente al KDC de **A**, conocido a partir de la dirección de correo de **A**. Cada petición lleva asociado un nivel de seguridad que indica el número de certificados de autenticidad de la clave que se desean. Por tanto, si una petición llega a un KDC con nivel de seguridad **n**, donde $n > 1$, éste KDC solicita a su superior una certificación con nivel de seguridad **n-1**, y así hasta que el nivel de seguridad sea 1. Cuanto mayor sea **n**, más seguridad tendremos de la autenticidad de la clave solicitada.



17. Una empresa de distribución de bebidas planea informatizar la venta directa en establecimientos de hostelería (bares, hoteles, etc.) mediante el uso de terminales autónomos en sus camiones de reparto. Los repartidores anotarán el pedido (productos y cantidades) de cada cliente usando los terminales. El terminal indica también si hay o no existencias en el camión para servir el pedido. El código del cliente se lee de una tarjeta que éste posee. El terminal emite entonces un recibo que es firmado por el cliente para confirmar la recepción de los productos. Por supuesto, el terminal almacena los datos del pedido. Además el repartidor entrega al cliente la factura correspondiente al último pedido para su cobro.

Una vez finaliza el reparto, el terminal se conecta a un ordenador central para descargar la información de los pedidos que se han servido. Esta información se utiliza para producir una factura para el cliente (que le será entregada y cobrada en la próxima visita del repartidor) y además se anota para utilizarla como referencia en próximos repartos.

Cuando un camión va a realizar el reparto se introduce la zona de destino y el ordenador imprime una orden de carga con una proporción de productos que se calcula en base a los pedidos anteriores en dicha zona. Además el terminal de ese camión se conecta al ordenador para inicializar las existencias del camión.

18. La empresa CD World se dedica a la venta de CDs a través de Internet. CD World utiliza un server al cual acceden los clientes con un browser. Además, para poder acceder a los servicios de esta empresa es necesario obtener una autorización de otra empresa (WebTrans) que dispone de medios para realizar cobros electrónicos seguros para el cliente (su sistema hace imposible que alguien descubra el nº de la tarjeta o la clave del cliente. Los procesos que se realizan en el sistema son:

Browser	<ul style="list-style-type: none"> ? Iniciar comunicación con CD World. ? Solicitar autorización a WebTrans. ? Una vez se dispone de la autorización, ésta se envía al server. ? Mientras se espera la autorización se puede ir viendo el catálogo y seleccionando CDs, que pasan a una “Bolsa de la compra” en el browser. ? Cuando el cliente lo indica, se pasa a realizar la compra real. En ese momento se muestra el contenido de la “Bolsa de la compra” y es posible seleccionar algunos de los CDs para hacer el pedido.
Server	<ul style="list-style-type: none"> ? Registrar contacto. ? Mostrar el catálogo para que el cliente vaya llenando su “Bolsa de la compra”. ? No se permite al cliente realizar la compra real hasta que se reciba la autorización. ? Cuando se reciba el pedido, aceptar o rechazar la venta según la autorización. ? Enviar a WebTrans la orden de pago del importe de la compra junto con la autorización.
WebTrans (No es parte de nuestro sistema)	<ul style="list-style-type: none"> ? Cuando se recibe una solicitud de autorización de un cliente para realizar compras a una empresa, se comprueban sus datos y se genera una autorización por el saldo disponible en su tarjeta. ? Cuando se recibe una orden de pago se comprueba la veracidad de esta y se lleva a cabo.

19. La memoria principal (MP) de un cierto ordenador está dividida en páginas de 512 bytes. Estas páginas pueden ser accedidas solicitando su lectura o escritura al controlador de MP, operaciones en las que se emplean 8 y de 21 a 25 ciclos de reloj, respectivamente. El ordenador dispone además de una memoria cache (MC) de alta velocidad que se inserta entre los procesadores y la memoria

principal. La MC está dividida en marcos de página, cada uno de los cuales puede contener una página de MP. El funcionamiento del sistema de memoria es el siguiente:

Cuando un procesador necesita acceder a una página de memoria, bien sea para escribir o leer un dato de ella, solicita dicha operación al controlador de la cache, que determina si dicha página está o no en la MC consultando una tabla asociativa que transforma direcciones de página de MP en direcciones de marcos de página de MC.

Si la página se encuentra en MC se realiza la operación de lectura o escritura, que dura entre 2 y 3 ciclos de reloj, finalizando así el ciclo de memoria. Si, por el contrario se produce un fallo de cache (la página solicitada no está en MC), la página debe ser cargada en la MC antes de realizar la operación de lectura o escritura. La página de MP se colocará en un marco de MC que se encuentre libre, o si no hay marcos libres, en el marco de MC que haya sido accedido menos recientemente. En el caso de que la página que se va a eliminar de MC haya sido modificada por algún proceso, ésta debe ser escrita en MP antes de ser reemplazada por la nueva página.

20.- *Un sistema doméstico de calefacción mantiene la temperatura de las habitaciones durante el invierno, de forma que la temperatura de cada habitación puede controlarse de forma relativamente independiente. El calor es solicitado desde cada habitación basándose en las temperaturas reales y deseadas para cada una de ellas. Cuando una o más habitaciones precisan calor, la caldera se enciende. Cuando ésta alcanza una temperatura suficientemente alta, se enciende un ventilador en la caldera para enviar aire caliente a las habitaciones a través de una serie de conductos. En el conducto de aire de cada habitación existe una válvula, controlada por el sistema para proporcionar calor sólo a aquéllas habitaciones que lo precisen. Cuando las habitaciones no precisan más calor, la caldera se apaga, pero el ventilador sigue en marcha, enviando calor hasta que la caldera se haya enfriado. Si la temperatura de la caldera excede un cierto valor de seguridad, la caldera se apaga, mientras que el ventilador sigue en funcionamiento.*