

Práctica complementaria de Máquinas de Estados en UML

Ejercicio 1. Un semáforo se caracteriza por su color actual (Rojo, Amarillo o Verde), y puede estar “Activo” o “Intermitente” (fuera de servicio). Cuando está “Activo”, muestra una luz roja durante 45 segundos, y a continuación una luz amarilla durante 3 segundos, seguida de una luz verde que dura 15 segundos. Luego vuelve a mostrar la luz amarilla por 3 segundos, y el ciclo vuelve a comenzar.

El semáforo se inicia normalmente estando “Activo” y en Rojo. Permanece “Intermitente” entre las 00:00h y las 06:00h de la madrugada. Y luego vuelve a activarse automáticamente.

- Realice un diagrama de clases conceptuales en UML.
- Modele una máquina de estados en UML.
- ¿Qué cambiaría si, además, el semáforo pudiera activarse y desactivarse manualmente, en cualquier momento?
- ¿Cómo modificaría la máquina de estados para que, al activarse el semáforo, retome su funcionamiento en el mismo estado en el que se encontraba cuando fue desactivado?
- e)

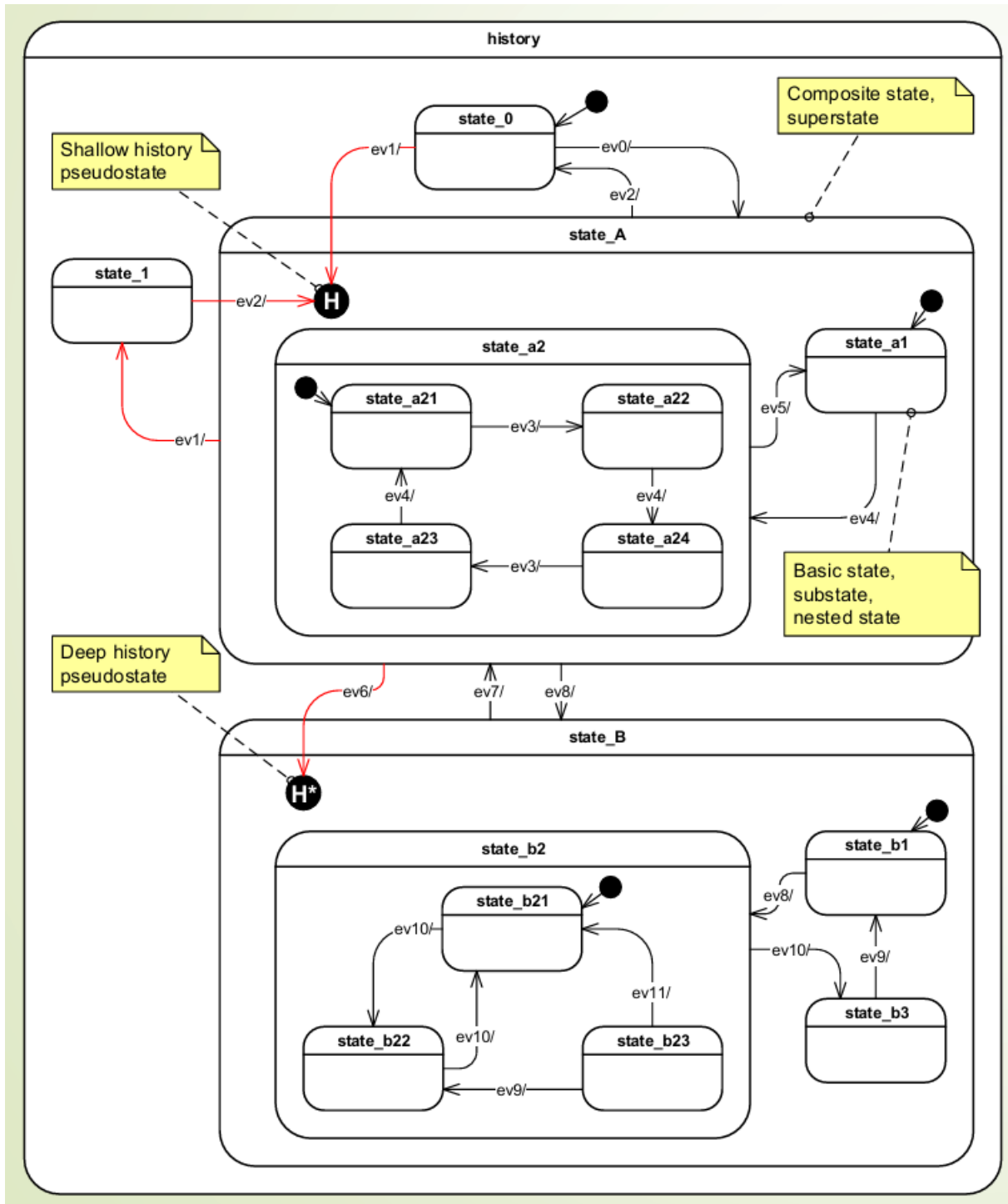
Ejercicio 2. Un aire acondicionado se inicia normalmente estando “Apagado”. Cuando se ejecuta el comando *encender()*, el dispositivo pasa a estar “Encendido”, y **normalmente** comienza “Monitoreando” de manera continua la temperatura del ambiente, y con una temperatura deseada de 24°C (la cual puede modificarse según las preferencias). Al detectar que la temperatura deseada es menor que la temperatura del ambiente, empieza a expulsar aire frío y pasa a estar “Enfriando”. Pero si la temperatura deseada es mayor que la ambiental, empieza a expulsar aire caliente y pasa a estar “Calentando”. Al detectar que la temperatura ambiente es igual a la deseada, vuelve a la situación inicial (“Monitoreando”).

Cuando se ejecuta el comando *apagar()*, el aire pasa a estar “Apagado”, y deja de monitorear la temperatura, de calentar o de enfriar, hasta que se vuelva a encender.

Cuando se ejecuta el comando *pausar()*, el aire pasa a estar “Pausado”, e interrumpe su funcionamiento. Si se ejecuta nuevamente el mismo comando, debe retomar su funcionamiento **exactamente en la misma situación** en la que fue pausado.

- Realice un diagrama de clases conceptuales en UML.
- Modele una máquina de estados en UML.

Ejercicio 3. Dado el siguiente diagrama, responda las preguntas planteadas a continuación:



- Si el estado activo es **a24** y ocurre el evento **ev1** seguido de **ev2**, ¿cuál es el próximo estado activo?
- Si el estado activo es **b23**, y ocurre el evento **ev7** seguido de **ev6**, ¿cuál es el próximo estado activo? En el mismo contexto, si acontece **ev7** seguido de **ev8**, ¿cuál es el estado activo final?

Ejercicio 4. Un ascensor conoce en todo momento su piso actual y el piso destino. Normalmente se inicia estando “Operativo”, y más específicamente “En Reposo”. Cuando se ingresa el piso destino, el ascensor sigue estando “Operativo”, pero específicamente puede estar “Subiendo” o “Bajando”, según el piso destino que se ingrese. Considere que el ascensor sabe responder al mensaje *subirUnPiso()* y *bajarUnPiso()*.

Además, cuando el ascensor sube puede estar “Subiendo lento” o “Subiendo rápido”. Y cuando está bajando, puede estar “Bajando lento” o “Bajando rápido”. Normalmente, la subida o la bajada se inician con velocidad lenta, y ésta puede modificarse mientras el ascensor se está moviendo.

Si se pausa el ascensor, deja de estar operativo para estar “Pausado”. Y al retomar su funcionamiento, debe hacerlo exactamente en la misma situación en la que estaba al ser pausado.

El ascensor puede desactivarse manualmente para hacerle reparaciones. En ese caso, permanece en estado “Fuera de servicio” hasta que es activado y vuelve a estar “Operativo” (se inicia normalmente).

- a) Realice un diagrama de clases conceptuales en UML.
- b) Modele una máquina de estados en UML.

Ejercicio 5. Juego de la caverna.



Se trata de un explorador que debe ir recorriendo una caverna, recolectando piedras preciosas. En el camino es atacado por murciélagos, a los que puede eliminar con un arma que lleva con él. Al llegar al final, debe abrir un pasadizo secreto, y entonces gana el juego.

El explorador tiene un estado de salud (inicialmente 100%), una cantidad de vidas disponibles (inicialmente son 3), una cantidad de diamantes recolectados y las coordenadas (x,y) de su posición en la caverna. El pasadizo se encuentra en la posición (0,0), y se abre si se para en una roca ubicada en ese punto, solamente si ya ha recolectado al menos 10 diamantes.

Mientras está jugando, el explorador puede estar juntando un diamante, usando su arma, caminando, o abriendo el pasadizo.

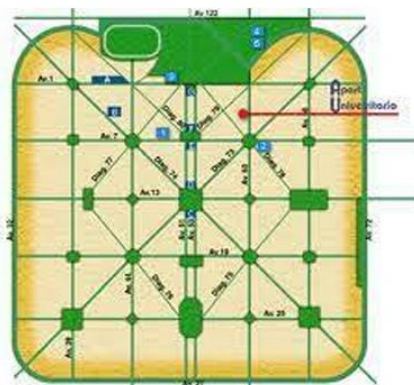
Cuando un murciélago muerde al explorador, su salud disminuye un porcentaje que depende de la ferocidad de este. Al llegar al 0%, muere (pierde una vida). Si le quedan vidas disponibles, vuelve al comienzo. Y si no, pierde el juego.

Se puede pausar la ejecución y, al continuar, el explorador debe reanudar el juego en la situación exacta en la que estaba al pausarlo. Si la pausa se extiende por 10 minutos, se pierde el juego.

La caverna tiene un grado de dificultad (entero) que se elige al principio del juego.

- a) Realice un diagrama de clases conceptuales en UML.
- b) Modele una máquina de estados en UML.

Ejercicio 6. La ONG “Platero” se dedica a la realización de proyectos propuestos por los ciudadanos de La Plata, para mejorar la calidad de vida en dicho municipio.



Cada proyecto que se crea, al principio se considera “propuesto”. Todos los proyectos tienen un nombre, un código de identificación, año de inicio y un grado de interés y las actividades que deben llevarse a cabo en el mismo. Las actividades tienen un nombre, una descripción, una fecha de fin, y un estado (planificada, en ejecución, finalizada).

Además, de cada proyecto se conoce quiénes son los miembros de la ONG que lo ejecutan (apellido, nombre, dni) y la función que cumplen en dicho proyecto (director, planificador o ejecutante).

Cuando se planifica un proyecto propuesto, se establece la fecha de fin de todas sus actividades pendientes, y el proyecto pasa a considerarse “planificado”.

Cuando se registra el inicio de la primera actividad de un proyecto planificado, éste pasa a considerarse “en ejecución”. Al registrarse el fin de la última actividad pendiente (no finalizada), el proyecto cambia su situación a “terminado”.

Un proyecto se considera “vigente” cuando está propuesto, planificado o en ejecución. Al comienzo de cada año, todos los proyectos vuelven a su situación inicial: propuesto. Salvo que hayan transcurrido más de 3 años desde su inicio. En este último caso, pasa a considerarse “terminado”.

Cada 15 días se registran como “suspendidos” todos aquellos proyectos que no tengan ninguna actividad “en ejecución”. Y un proyecto suspendido, puede reanudarse en la situación en la que quedó, para lo cual debe proveerse una operación que lo haga explícitamente.

Si el grado de interés (que es un número real entre 0 y 1) llega a ser menor a 0.3, el proyecto es “cancelado”. Esto puede ocurrir por diversas causas.

- a) Realice un diagrama de clases conceptuales en UML.
- b) Modele una máquina de estados en UML.