



**Instituto Politecnico Nacional**



**ESCOM “ESCUELA SUPERIOR DE CÓMPUTO”**

*ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS*

*ELEMENTOS Y PROPÓSITOS DE LOS DIAGRAMAS DE ESTADOS*

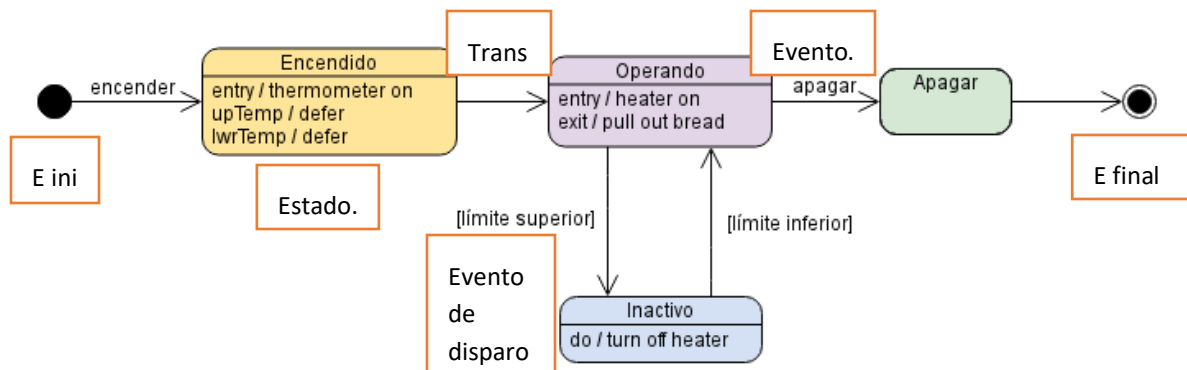
PROFA: Reyna Melara Abarca

ALUMMNO:

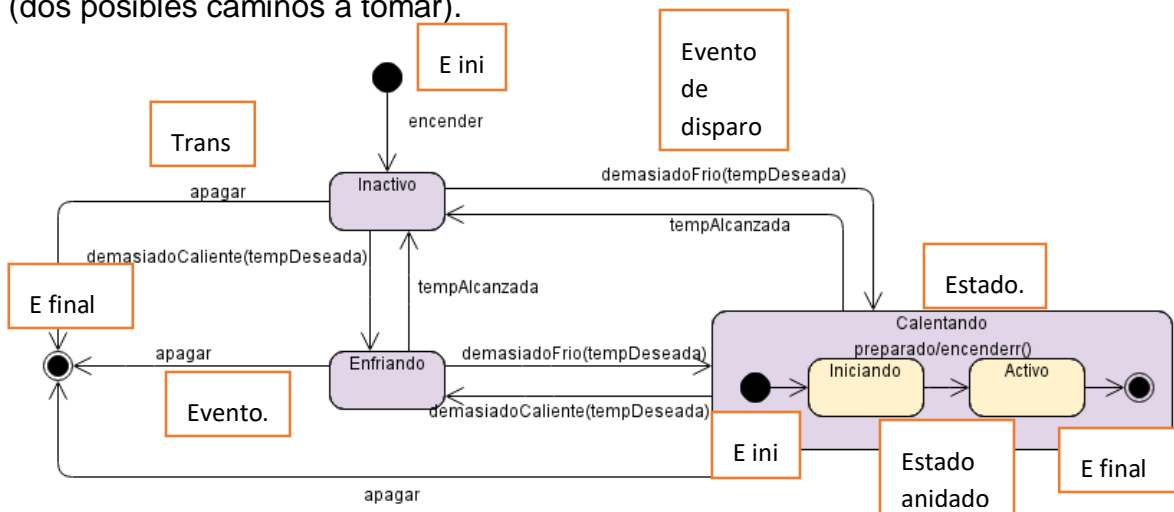
Rojas Alvarado Luis Enrique

GRUPO: 2CM9

# 1. Identifique los elementos de los diagramas de estado y describa brevemente lo que interpreta de cada diagrama.

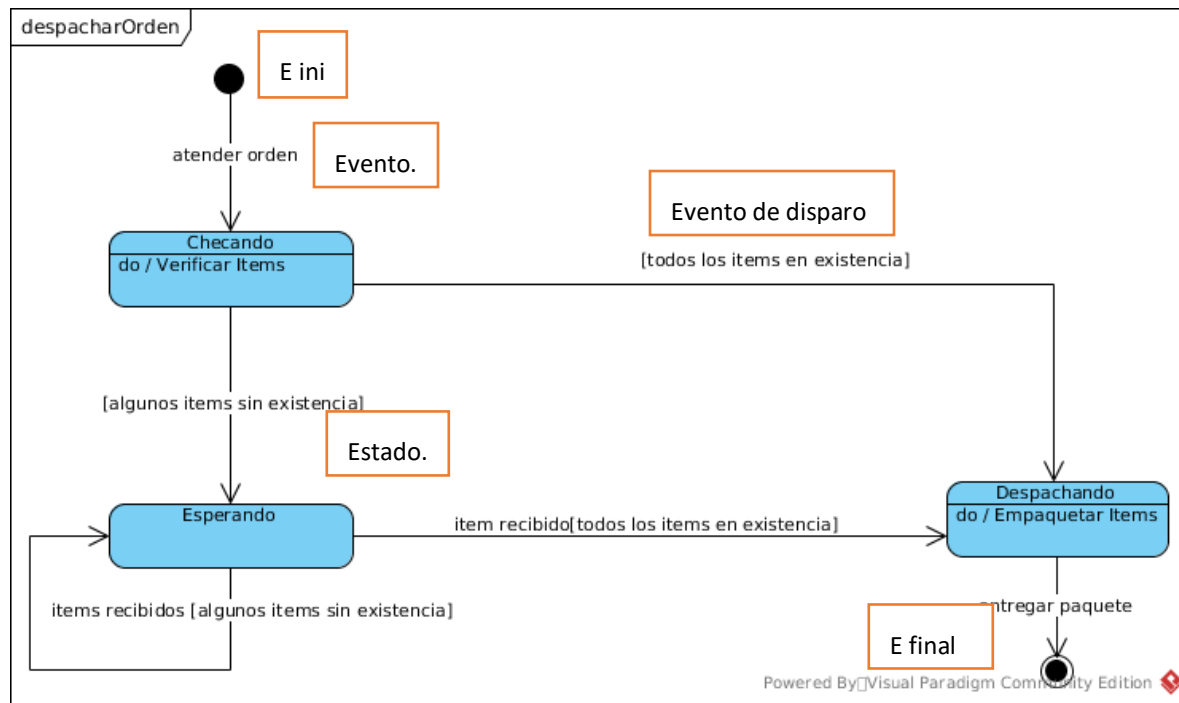


En éste diagrama se puede entender que es el sistema de un calentador, dónde el estado inicial es encendido: éste estado el termómetro se enciende, puede subir la temperatura, también puede postergar el tiempo de calentado. Pasando al estado operando el calentador se enciende a la entrada y a la salida saca el pan. De ahí tiene dos caminos posibles: puede ir al estado de apagar directamente y apagar el calentador o mandar como evento el límite superior y entrar en estado inactivo donde se apaga el calentador, mandando como respuesta un límite inferior (evento) donde regresa al estado operando y repitiendo el mismo ciclo anterior mencionado (dos posibles caminos a tomar).

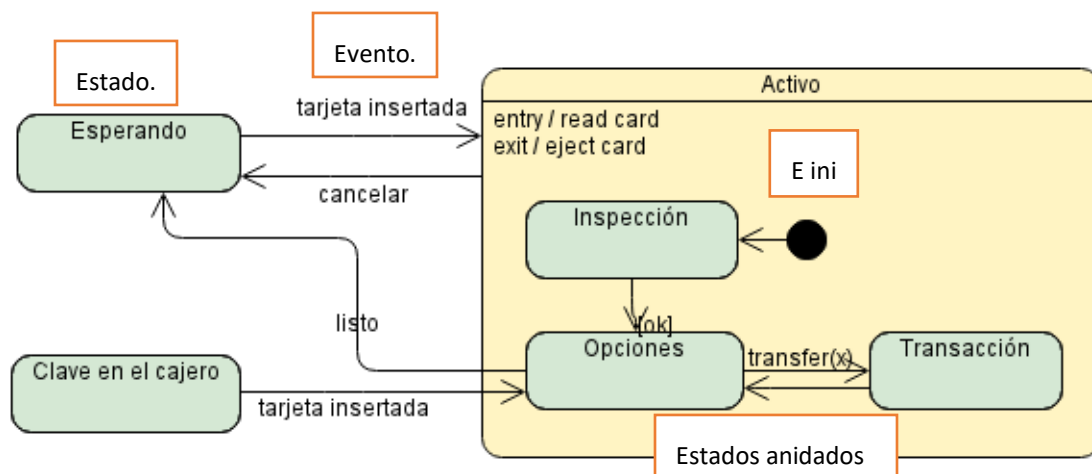


Éste diagrama de estados es de igual manera de un calentador. El cual su estado inicial tiene un evento disparador que con una transición llega al estado inactivo, en donde tiene tres posibles caminos: uno es apagar directamente el calentador, el segundo manda como parámetro del evento **demasiadoFrio**, una temperatura deseada para entrar en una nueva máquina de estados con el nombre de calentando, que en después de su estado inicial entra a un estado iniciando, manda como evento **preparado/encender** para llegar al estado activo y posteriormente llega al estado final y devuelve un evento **demasiadoCaliente** con los argumentos

Temperatura deseada al estado enfriando, que también es el tercer camino que se puede llegar desde inactivo. Y directamente saliendo de esta sub máquina de estados se puede apagar el calentador. El estado Enfriando puede regresar un evento al estado que contiene una sub-máquina de estados llamado demasiado frío con los argumentos TempDeseada y repetir el ciclo anteriormente mencionado, o simplemente apagar el calentador directamente.



En ésta máquina de estados representa un despachador, en el cual el evento disparador del estado checando es atender orden, en dónde verificará los ítems, si todos los ítem son existentes pasa al estado de despachando donde empaqueta los ítems, si no están en existencia entra al estado de esperando donde los ítems recibidos solo hay algunos sin existencia regresa al estado esperando y los ítems en existencia los pasa al estado despachando para posteriormente entregar el paquete y pasar al estado final de la máquina de estados del despachador.



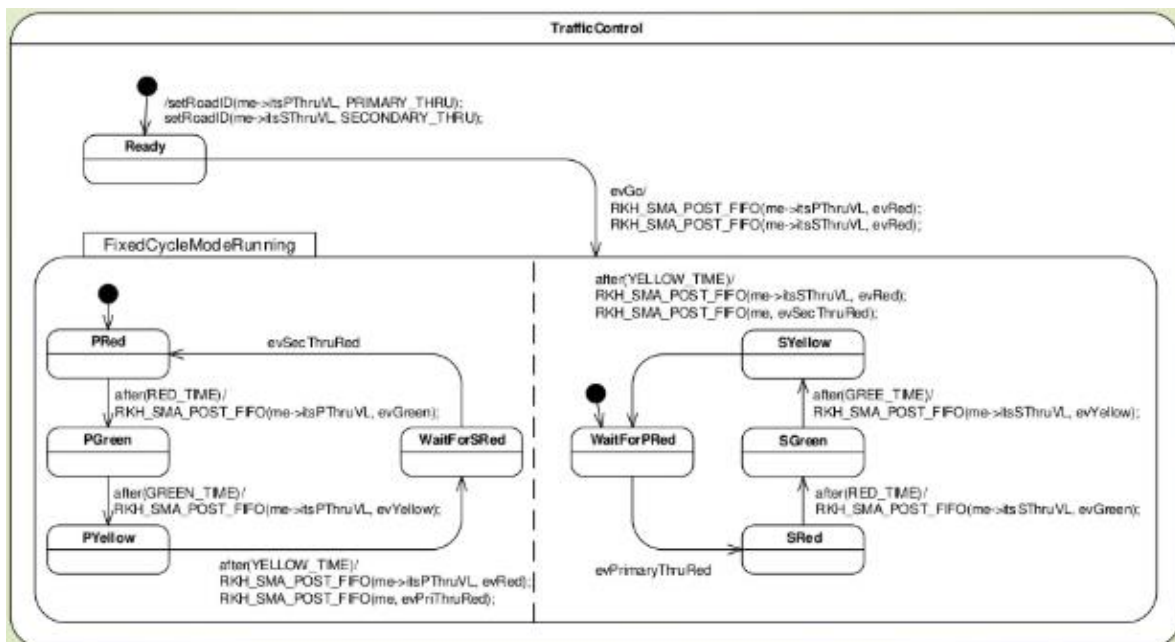
Esta máquina de estados simula un cajero automático, donde no tiene estado final. Para su estado inicial, que está dentro de un estado activo (siempre) pasa a una inspección si da ok pasa a opciones en donde se puede pasar con transferir a una transacción y regresar o a un estado de esperando la tarjeta insertada y vuelve a entrar en el estado activo donde leerá la tarjeta y al salir saca la tarjeta. Con el evento cancelar regresa al estado esperando y se forma un ciclo. Si se ingresa por clave en el cajero la transición contendrá el evento tarjeta insertada, donde regresará al estado de opciones.

## 2. ¿Cuándo pueden ser útiles los diagramas de estado bien estructurados en el modelado de sistemas de software usando UML?

Pueden ser útiles cuando se quiere hacer una representación del funcionamiento del sistema y para darle a conocer a un diseñador como es que el sistema va a funcionar en forma general.

## 3. ¿Qué es un estado ortogonal? realice un diagrama en el que se pueda observar.

Un estado ortogonal es un estado compuesto y contiene más de una región.



## 4. ¿Qué es un estado historia, cómo se representa?

Se asocia con las regiones de los estados compuestos. Mediante el cual, una región mantiene un registro de la última configuración de estados activa. Permite, si se

requiere, volver a esa misma configuración de estados: La próxima vez que la región esté activa, o si hay una transición que termina en su historia.

Existen 2 tipos de historia: Shallow, que mantiene registro sólo del último sub-estado directo visitado de la región. Deep que mantiene el registro de la configuración de estados de la región.

## **5. ¿Qué son los efectos de entrada/salida?**

Al entrar a un estado, se ejecuta, luego el comportamiento efecto asociado con la transición entrante: Su comportamiento de entrada, si está definido y finalizado el mismo, comienza inmediatamente el comportamiento de actividad, si está definido. Cuando se entra a un estado compuesto simple: Entrada por defecto, por shallow history, por Deep history, por entry point, o por es una entrada a un estado ortogonal, entrar desde el borde (por defecto) o entrar explícitamente a una o más regiones.

Al salir de un estado, ya sea compuesto o simple, se ejecuta su comportamiento de salida. Interrumpiendo, previamente, su comportamiento de actividad, si es que existe. Al salir de un estado compuesto, la salida comienza desde el estado más interno en la configuración de estados activa, ascendiendo por jerarquía. Si la salida ocurre desde un exitPoint, se ejecuta el exit luego del effect de la transición que termina en dicho pseudoestado. Al salir de un estado ortogonal, se abandonan todas sus regiones, luego de ejecutar el exit de dicho estado.

## **6. ¿Qué es una transición interna?**

Es un tipo especial de transición local (que sale de su vértice de origen), cuyo estado origen coincide con su estado destino, también conocida como auto-transición, donde el estado origen nunca se abandona y por lo tanto, cuando se ejecuta, no se efectúan los comportamientos entry y exit. Este tipo de transición puede definirse únicamente si el vértice origen es un estado.

## **7. ¿Qué es una actividad-DO?**

Si está definido, comienza su ejecución cuando se ingresa al estado, finalizado su comportamiento entry. Se ejecuta concurrentemente con otros comportamientos del mismo estado, hasta que: la actividad termine, en cuyo caso se genera un evento de terminación. O se abandone el estado, interrumpiendo así la actividad.

**8. ¿Pueden modelarse subestados concurrentes usando los diagramas de máquinas de estado? Explique y en su caso proporcione un diagrama de ejemplo.**

Se les llama submáquina, son un medio por el cual la única especificación de SM puede reutilizarse más de una vez.

