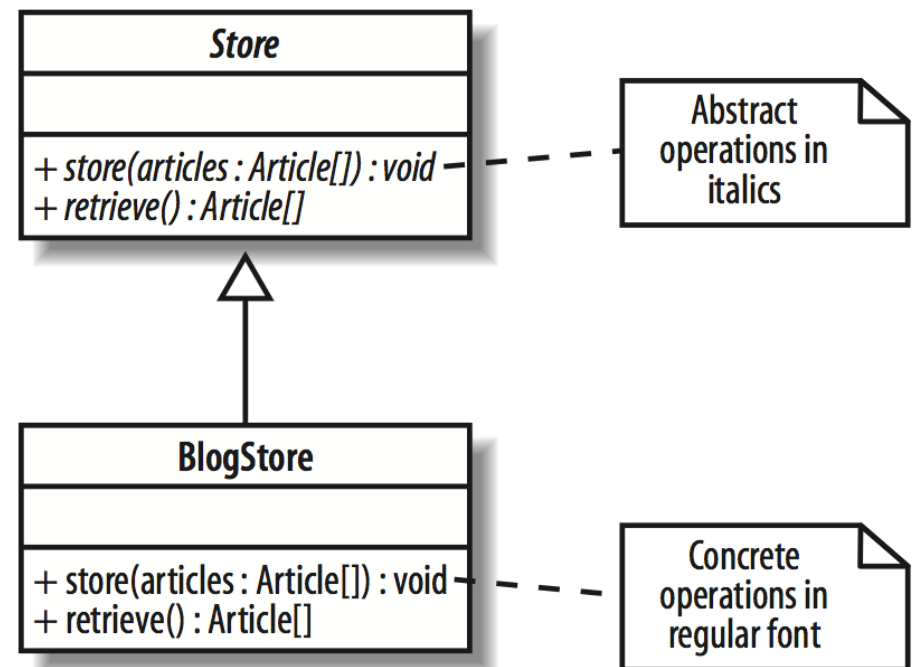


Clases Abstractas

- Están destinadas a ser especializadas (clases concretas)
- Operaciones se escriben en cursiva



Interfaces

- Colección de operaciones (métodos) que las clases concretas deben implementar
- Similar en uso a clases abstractas
- Como no son clases no hay problema en que muchas clases implemente una misma interfaz
- Es bueno verlo como un contrato
- Si hay atributos son estáticos o constantes

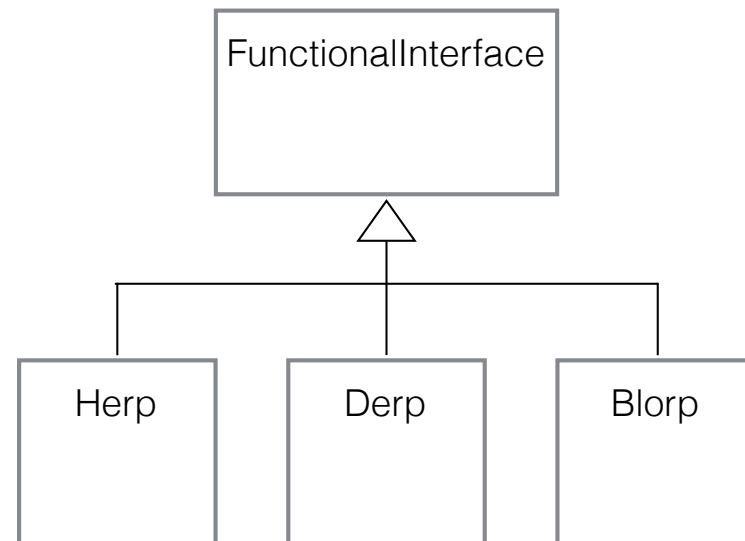
Clases Abstractas e Interfaces en Ruby

```
class FunctionalInterface
  def do_thing
    raise "This is not implemented!"
  end
end
```

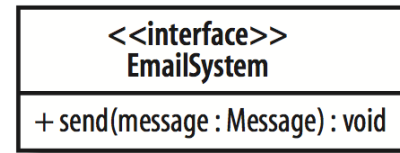
```
class Herp < FunctionalInterface
  def do_thing
    puts "herp"
  end
end
```

```
class Derp < FunctionalInterface
  def do_thing
    puts "derp"
  end
end
```

```
class Blorp < FunctionalInterface
  def do_some_other_thing
    puts "whoops"
  end
end
```



Interfaces en UML



Stereotype Notation

Or



"Ball" Notation

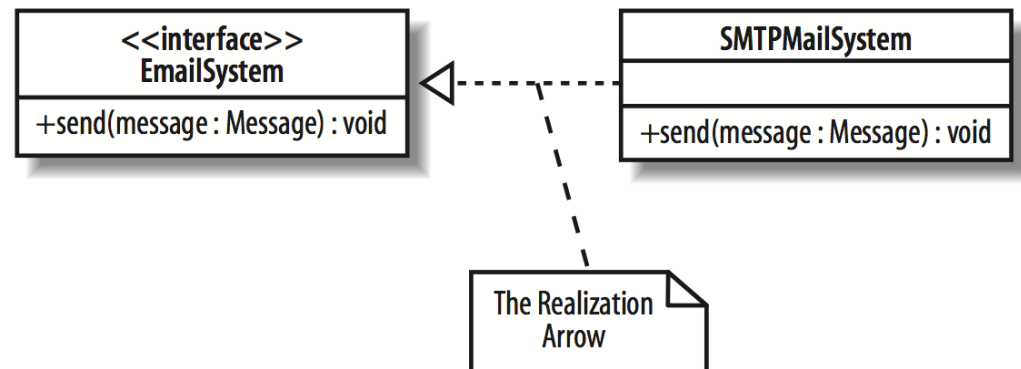
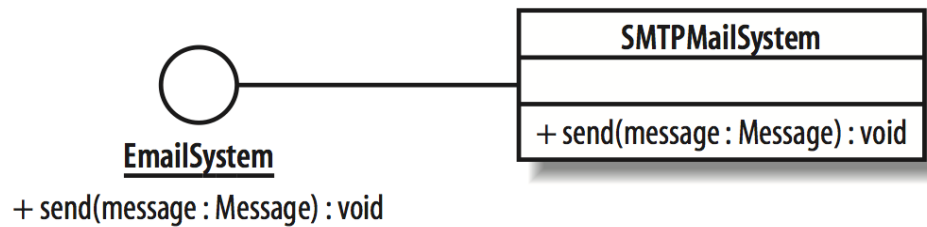
- Puede usarse un *estereotipo* o un símbolo especial

```

public interface EmailSystem {
    public void send(Message message);
}
  
```

```

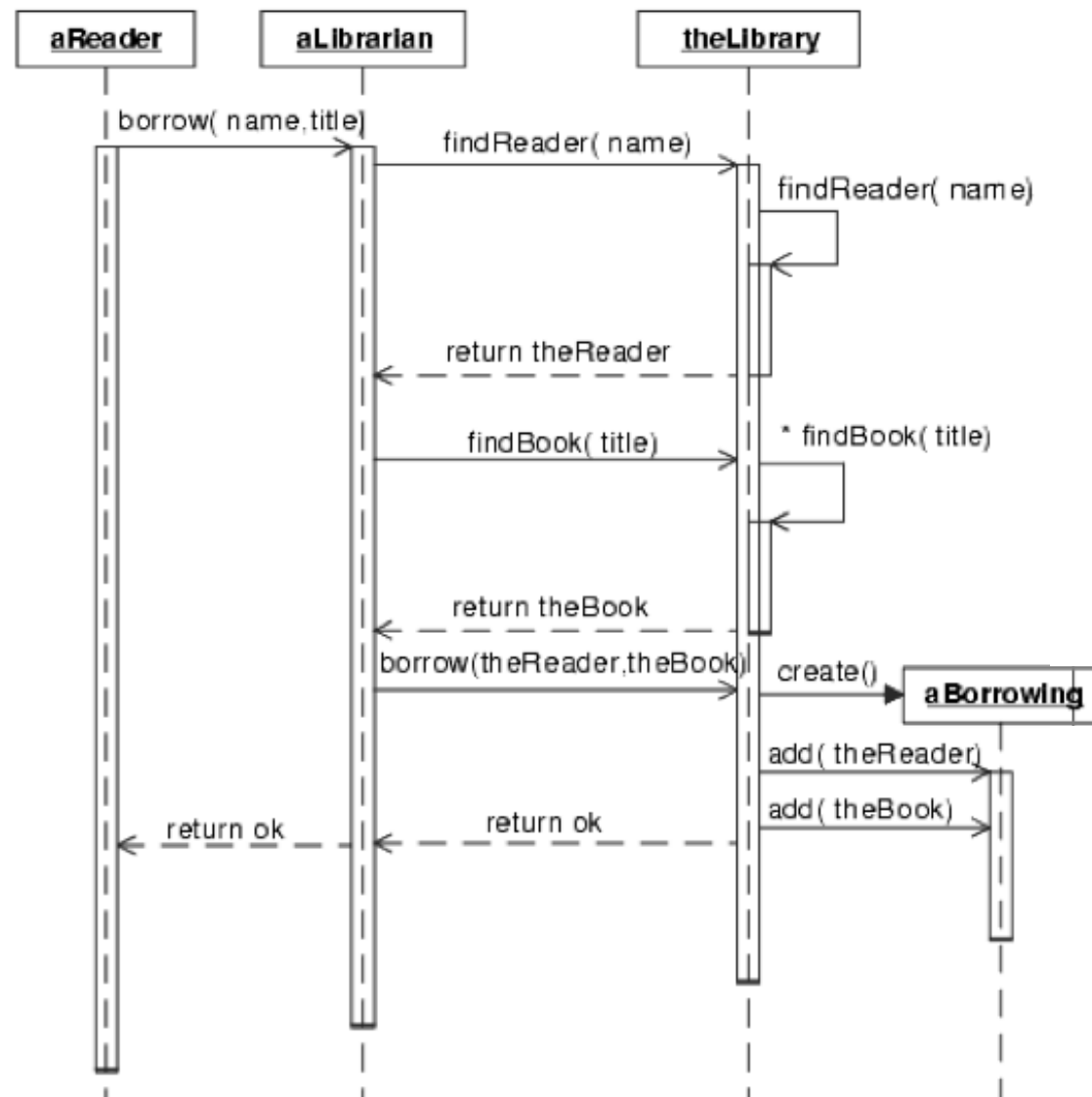
public class SMTPMailSystem implements EmailSystem {
    public void send(Message message)
    {
        // Implement the interactions to send the message
    }
    // ... Implementations of the other operations ...
}
  
```



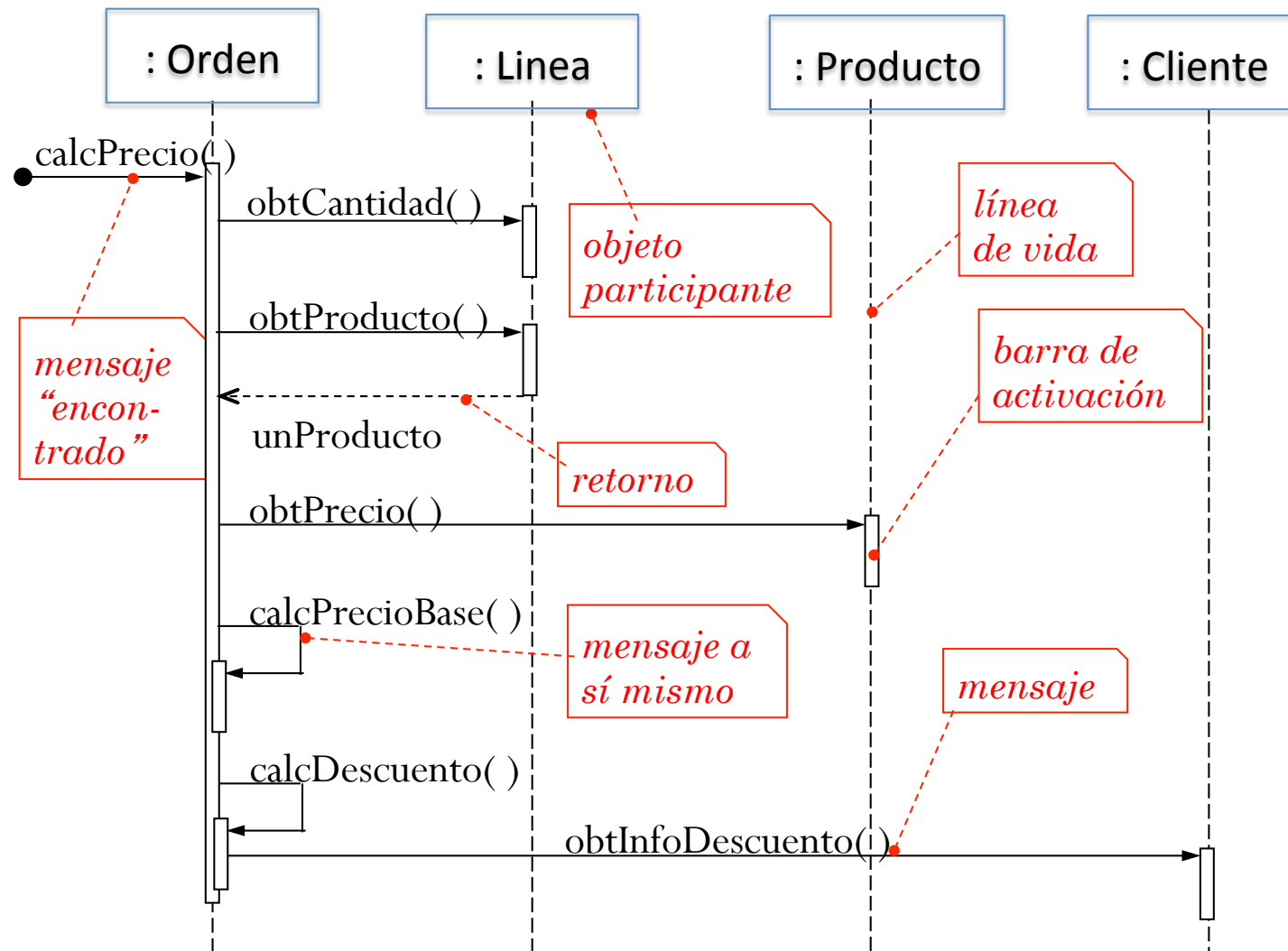
Describiendo la Interacción de los objetos con UML

- Diagramas de Secuencia (posiblemente el segundo más usado después del de clases)
- Lineas de tiempo (hacia abajo) para un conjunto de objetos que cooperan para realizar una operación
- Envío de mensajes (invocación de métodos) se representa por flechas horizontales

Ejemplo Diagrama de Secuencia

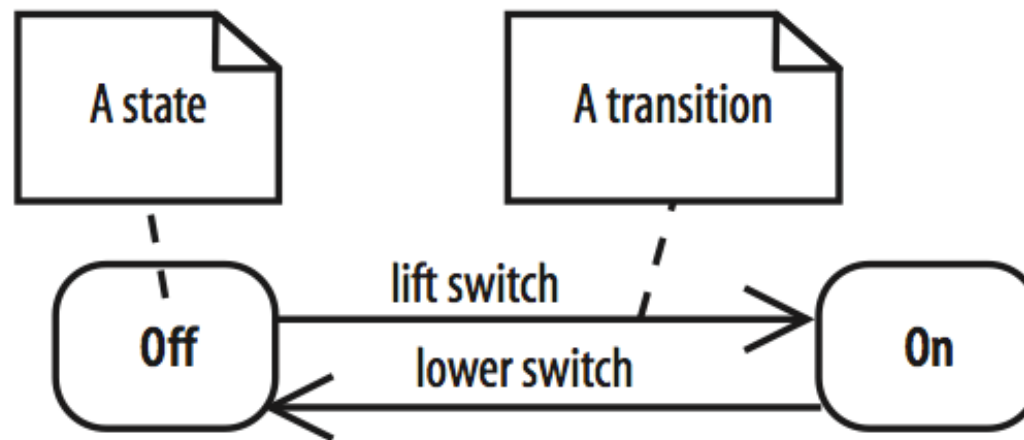


Conceptos en diagrama de Secuencia

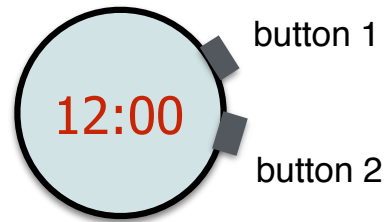


Diagramas de Estados (UML)

- Lo esencial : estados y transiciones entre estados

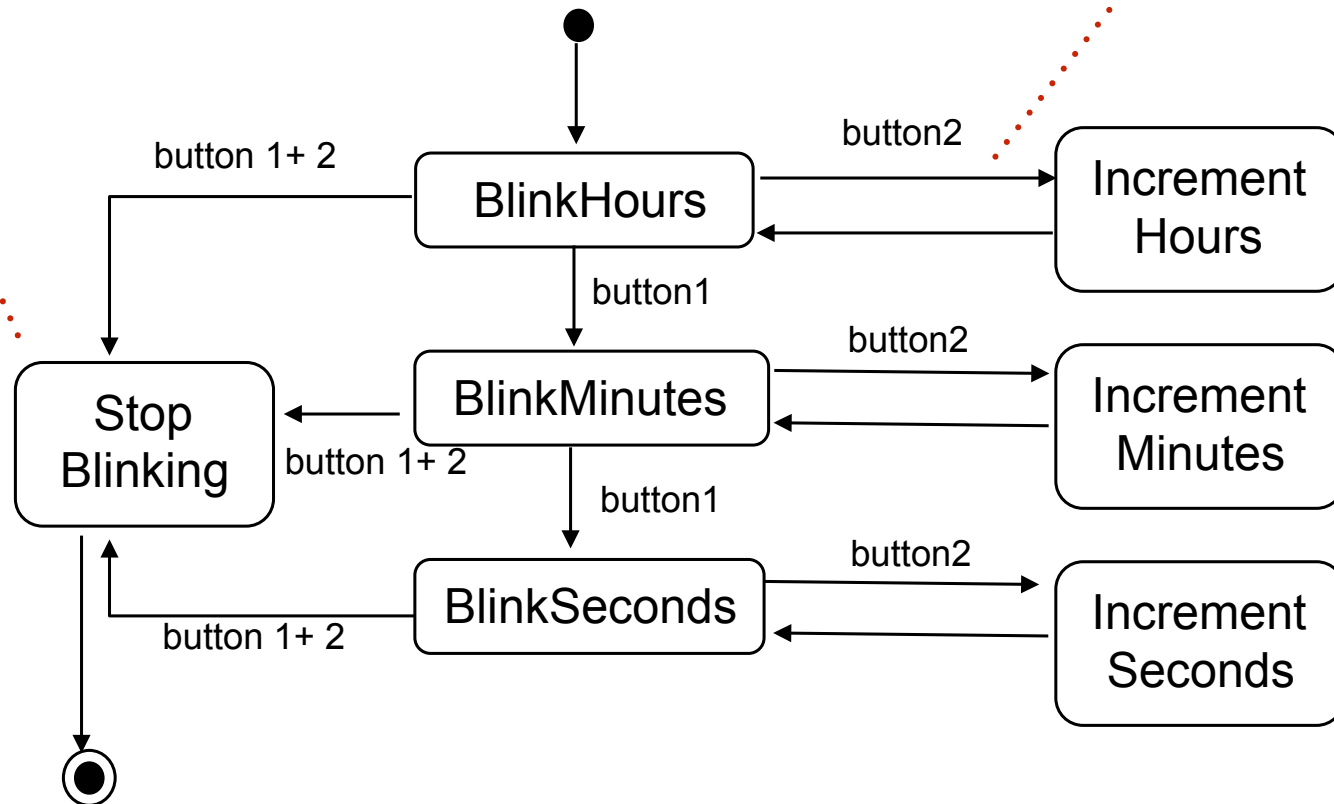


Ejemplo

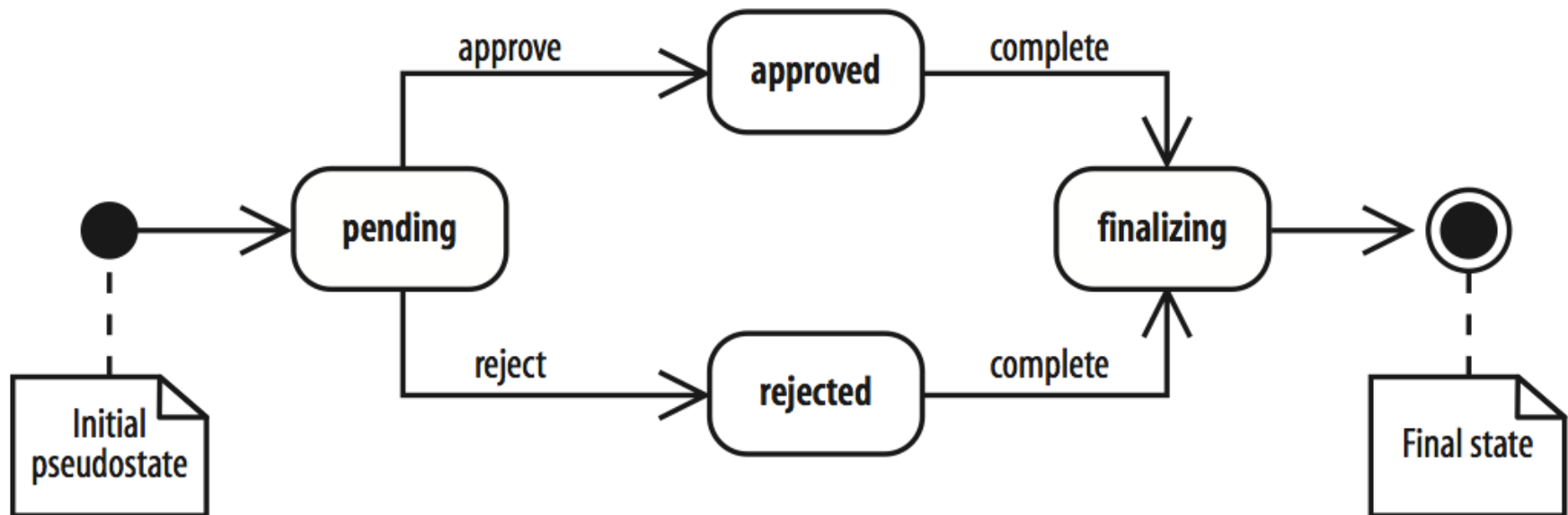


estado

transición

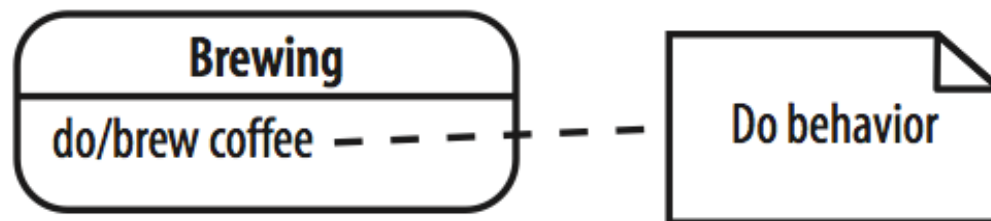


Estados Inicial y Final

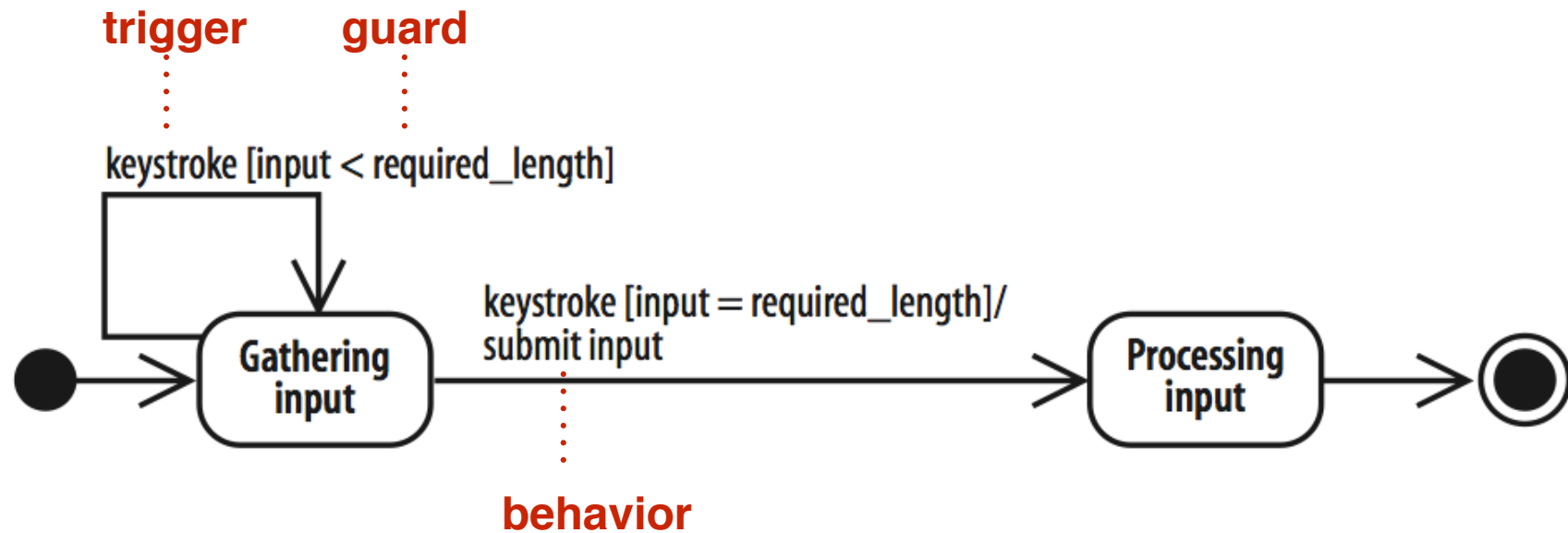


Estados Activos

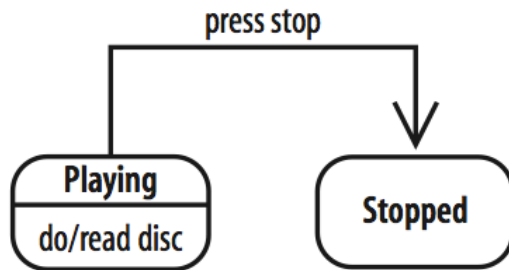
- Un estado pasivo es por ejemplo on/off
- Un estado activo es algo que está ocurriendo o haciéndose (por ejemplo “calculando”)
- En este caso se puede describir comportamiento en el interior del estado (parte tan pronto se ingresa al estado)



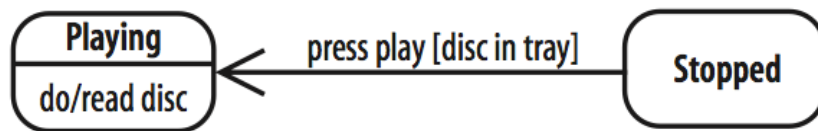
Transiciones



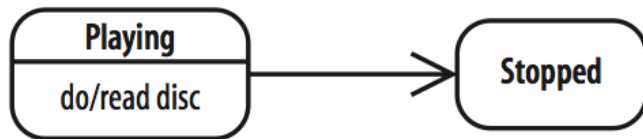
Variaciones



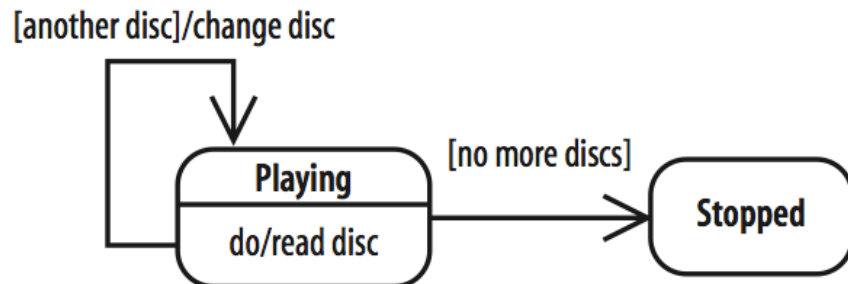
trigger simple



trigger con guard



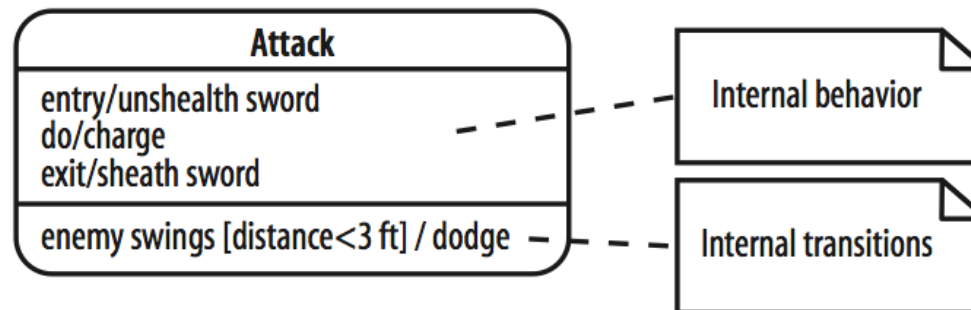
transición automática



guards para elección

Mas sobre comportamiento durante un estado

- aparte de “do” puede especificarse un “entry” y un “exit”
- entry behaviour - sucede tan pronto se activa estado
- exit behaviour - sucede inmediatamente antes de salir de ese estado
- transición interna - reacción dentro del estado



Ejercicio (Diagrama de Estados)

Debemos escribir el software de un pequeño *timer* de cocina como el que se muestra en la figura de la derecha. El aparato tiene 10 teclas con los números 0 al 9 mas una tecla *start* y una tecla *stop/reset*

Cualquier tecla numérica comienza el ingreso del tiempo de partida (0 a 59 minutos). Se puede ingresar uno o dos dígitos. Si se ingresan más números quedan los dos últimos. Las teclas numéricas solo pueden usarse si el *timer* está detenido y en cero.

El *start* comienza la cuenta regresiva que solo se detiene cuando el usuario presiona el *stop*. Sin embargo, al llegar a cero comienza a emitirse una alarma durante 1 minuto. minuto (el reloj continúa, pero ahora en aumento para dar cuenta del tiempo que ha transcurrido desde que se completó el tiempo inicial)

Al presionar *stop* la cuenta se detiene. Al presionar nuevamente *stop* los números vuelven a cero.

Si el timer está contando y se presiona *start*, comienza nuevamente desde el mismo valor inicial.

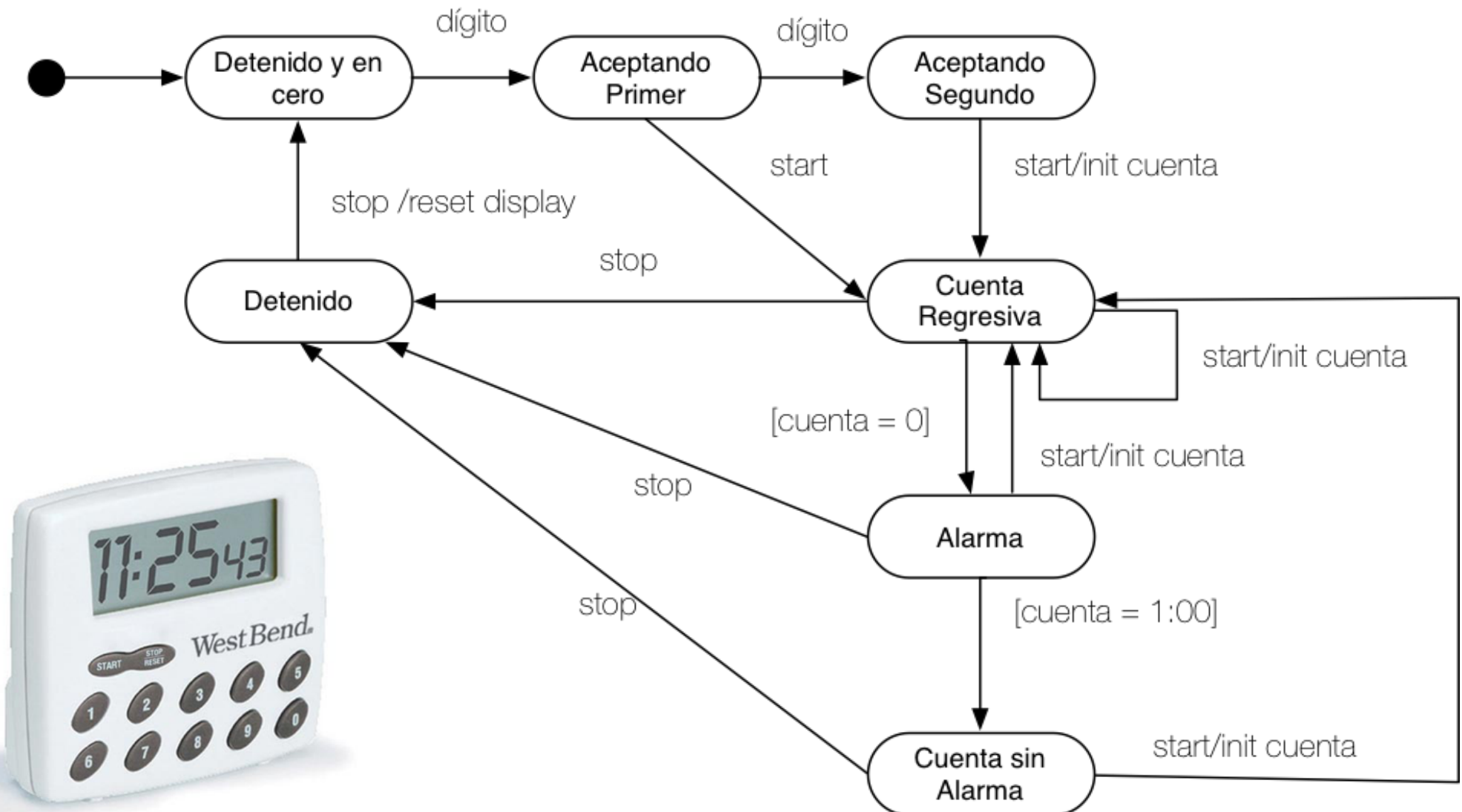
Modele el comportamiento mediante un diagrama de estados.



Solución

Las transiciones son ocasionadas por la acción de un botón salvo las de cuenta regresiva a alarma y de alarma a cuenta sin alarma que son espontáneas cuando se cumple la condición.

Hay 5 transiciones que incluyen una acción. En cuatro casos para inicializar el contador al valor ingresado con los dígitos y en el otro para resetear el display



Problema: Modelo Dominio

Una empresa constructora que mantiene simultáneamente varias obras en ejecución maneja un pool de máquinas comunes que son usadas en las distintas obras. La empresa tiene además un pool de ingenieros que son asignados temporalmente a una obra dada.

Cada una de las máquinas tiene un nombre y un identificador además de otra información asociada: características, costo de operación por hora, horas de uso, horas desde última mantención, etc. Una máquina puede ser operada por un cierto número de operadores que la empresa emplea en una modalidad por hora y para cada una de ellas se mantiene una historia con el detalle de las mantenciones que se le han realizado.

Las mantenciones son realizadas por ingenieros de mantención de la empresa constructora y para realizarlas se apoyan en manuales de servicio asociados a cada máquina (puede haber varios para cada una de ellas). En una mantención puede requerirse un número de repuestos.

Cada obra tiene una planificación (carta gantt) que lista todos los trabajos a realizar. Algunos de ellos tienen asociados la participación de máquinas y otros no. Una máquina puede estar asignada a más de un trabajo, pero siempre en la misma obra.

Cada fin de semana se debe pagar a cada operador de acuerdo al trabajo realizado para lo cual se emite un detalle de cada trabajo realizado en este lapso (obra, fecha, hora inicio, hora término, máquina asociada).

Se pide construir un modelo de dominio para este problema en forma de un diagrama de clases. Incluya en las clases los principales atributos (de dominio). Incluya las decoraciones y cardinalidades en las líneas de asociación

Solución

