



Práctica 5 - Patrón de Estados
Diseño de Arquitectónico de Patrones

Universidad de La Laguna

Francesco Marelli
(alu0101161730)

9 de noviembre de 2023

Indice

1. Introducción	2
2. Patrón de Estado	3
3. Descripción del programa y aplicación	4
4. Diagrama de Clase	5
5. Tabla de Tiempos	6

1 Introducción

En esta práctica se pretende crear una aplicación que simule el funcionamiento de los semáforos. Este semáforo también funcionará para personas nonvidentes, por lo que no sólo muestra luz sino que también en algun caso emite diferentes sonidos según el color. El semáforo funcionará de la siguiente manera: se pondrá en rojo durante 10 segundos, luego en naranja durante 3 segundos y finalmente en verde durante 10 segundos. La luz verde parpadeará más rápido y emitirá un sonido durante 3 segundos antes de cambiar a roja.

Se pretende diseñar este programa con aplicando el diseño del patrón de estados que se explicará a continuación. En el informe se incluyen también una breve descripción del programa, los diagramas UML y una tabla de los tiempos con las consideraciones y dificultades surgidas a lo largo de la realización de la práctica.

2 Patrón de Estado

El patrón de Estado es un patrón de diseño de comportamiento que permite a un objeto alterar su comportamiento cuando su estado interno cambia en tiempo de ejecución de forma sencilla y mantenible.

Para abordar este tipo de problemas, se propone implementar el patrón de diseño de Estado. La estructura de este patrón se puede visualizar en la Figura 2.1, donde el Contexto es el encargado de definir la interfaz para el cliente, manteniendo una instancia del Estado Concreto que representa el estado actual. La interfaz del Estado encapsula las responsabilidades asociadas con un estado específico del contexto.

Las subclases de Estado Concreto implementan el comportamiento correspondiente a cada estado definido en el contexto. Este enfoque modulariza y organiza el comportamiento relacionado con el estado, permitiendo que el contexto cambie de estado dinámicamente mientras mantiene un bajo acoplamiento entre las clases.

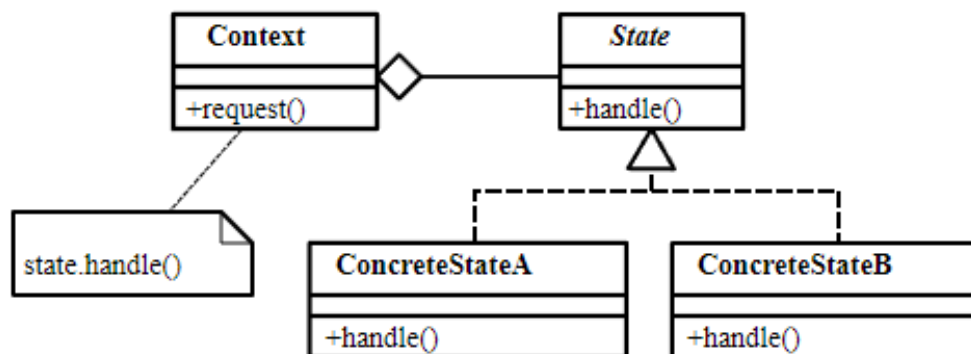


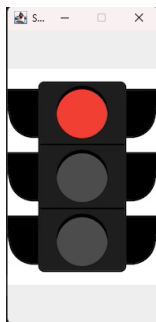
Figura 2.1: UML del Patrón de Estado

En este programa el Contexto se identifica con la clase Semaforo, la clase Estado como la interfaz Estado y las clases EstadoVerde, EstadoRojo, EstadoAmbar y EstadoVerdeParpadea son los estados concretos.

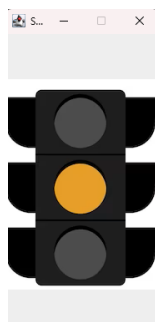
3 Descripción del programa y aplicación

El programa realiza una simulación del semáforo a través de una ventana. La ejecución empieza con el estado inicial, el rojo, que dura 10 segundos. Automaticamente pasará al estado ambar que dura 3 segundos. El siguiente estado es el verde que dura 7 segundos, luego pasará al estado verde intermitente otros 3 segundos. El programa seguirá funcionando en bucle. Para trabajar con los tiempos se ha hecho uso de un Timer en la clase SeamforoVentana y de Threads en el método de ejecución de cada estado. La interfaz estado es la que se encarga de comunicar con el Contexto que es la clase Semaforo. Los metodos principales son de acceso, además de ejecutarAcción, que explicita el comportamiento de cada estado concreto. Es el que se encarga de dictar el tiempo de ejecución de cada estado, reproducir su sonido, mostrar la luz de referencia y de cambiar al siguiente estado. En cada estado estos comportamientos son diferentes.

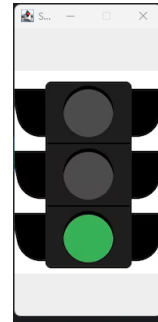
En la imagen 3.1 se puede apreciar la ejecución de del semáforo.



(a) Semaforo Rojo



(b) Semaforo Amarillo



(c) Semaforo en verde

Figura 3.1: Ejemplo de ejecución del programa

4 Diagrama de Clase

En el diagrama de clase en la figura 4.1 describe la estructura de las clases de la práctica. La clase Estado hace de Interfaz a los distintos estados concretos: Estado VerdeParpadea, EstadoAmbar, EstadoRojo, Estado Verde. El contexto Semaforo se ocupa de setear el estado actual, ejecutar la acción de cada estado y trabajar con la interfaz gráfica. SemaforoWindow se ocupa de Jcuntar todos los elementos y ejecutar el programa y representarlos en la interfaz gráfica.

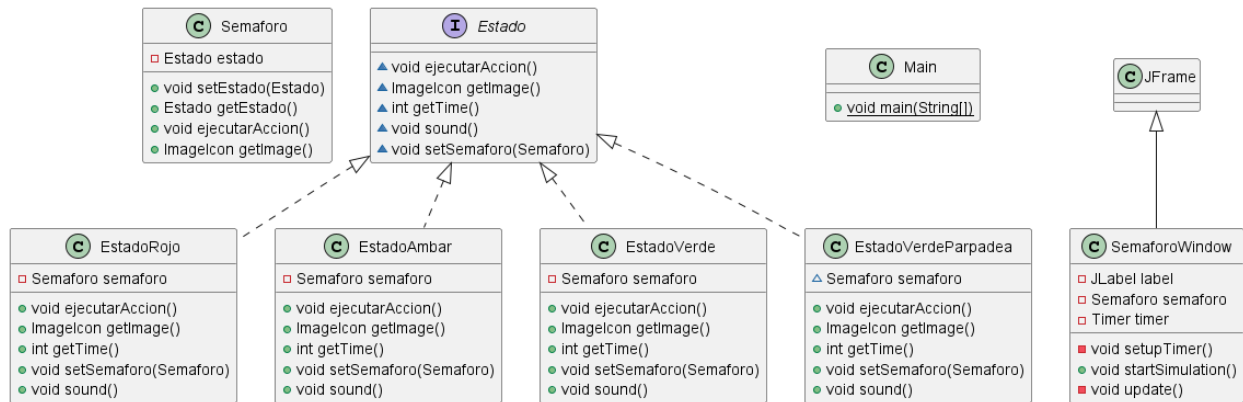


Figura 4.1: UML del Patrón de Estado

5 Tabla de Tiempos

En la tabla 5.1 se ilustra la organización de los tiempos y el tiempo efectivo de realización para cada tarea de la práctica. En general se ha tardado más de lo esperado. En principio de la práctica no me pareció difícil pero encontré muchos problemas en la resolución de la misma. Trabajar con ventanas especialmente aplicando hilos, imágenes y sonidos ha sido en mi opinión complicado, ya que era la primera vez que juntaba todos esos conceptos a la vez (Había ya trabajado con imágenes y ventanas solamente). Trabajar en Latex para el informe no ha sido un problema.

Tarea	Tiempo Estimado	Tiempo Efectivo
Estructura Patrón	30 min	1 h
Simulación	2 h	3 h 30 min
Informe	1 h	1 h
Total	3 h 30min	5 h 30

Cuadro 5.1: Tabla de tiempos