

Para desarrollar el proyecto del elevador, primero analicé cuáles eran los elementos fundamentales que intervienen en el funcionamiento de un ascensor real. A partir de eso, organicé la solución utilizando los principios de Programación Orientada a Objetos vistos en clase.

Identifiqué que necesitaba modelar: el elevador, la puerta, los botones internos y externos, la dirección de movimiento y un controlador que gestione todas las solicitudes. Para evitar código repetido decidí crear una clase abstracta llamada “Botón”, de la cual heredarán “BotonPiso” y “BotonInterno”, Cada botón implementa el método “presionar()” según su función: los externos envían solicitudes de piso y dirección, mientras que los internos sólo envían el piso deseado.

La clase “Elevador” mantiene el estado del ascensor, como el piso actual, la dirección y su puerta. No toma decisiones por sí mismo, sino que ejecuta las órdenes que recibe. La parte lógica principal está en la clase “ControlElevador”, la cual actúa como el “cerebro” del sistema. Allí decidí usar un “TreeSet” para almacenar las solicitudes, ya que mantiene los pisos ordenados y evita duplicados, lo que simplifica la búsqueda del piso más cercano cada vez que el elevador debe moverse.

El método “paso()” del controlador simula cada instante de funcionamiento: selecciona el piso más cercano, mueve el elevador un piso, abre la puerta si llega a destino y retira la solicitud atendida. Esto permite que el sistema funcione como una simulación sencilla pero realista de un elevador.