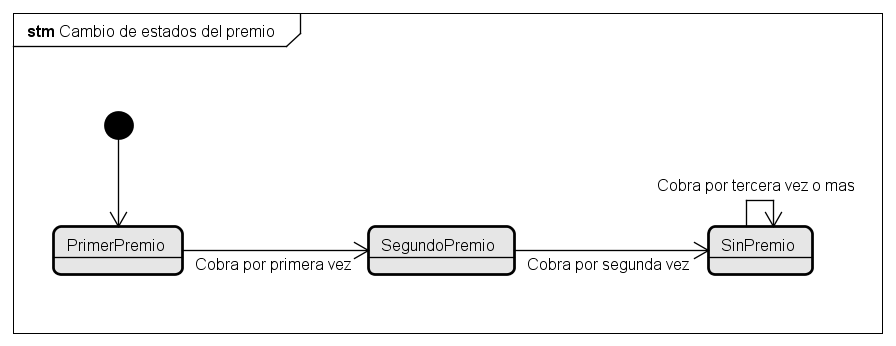
**Quini 6**

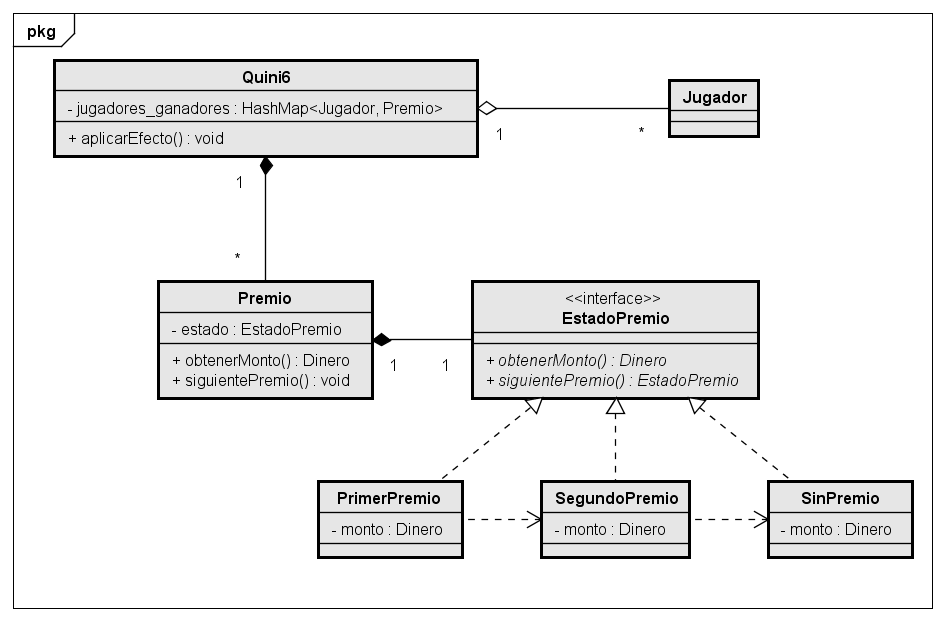
Para la solución a este requerimiento se utilizó el patrón “state” que permite que un objeto altere completamente su comportamiento en función de un cambio de estado de manera de comportarse como una clase completamente diferente.

Entonces, enfocándonos en el problema, necesitamos que el (casillero) Quini6 entregue un premio diferente según el jugador haya caído en el casillero una, dos o más veces.

Por lo tanto el objeto (casillero) Quini6 conoce a los jugadores que cayeron en él y le asigna un premio a cada uno de ellos. Si el jugador cae por primera vez el Premio se comportará como un *“PrimerPremio”*, si cae por segunda vez se comportara como un *“SegundoPremio”* y a la tercera vez se comportará como un *“SinPremio”*.

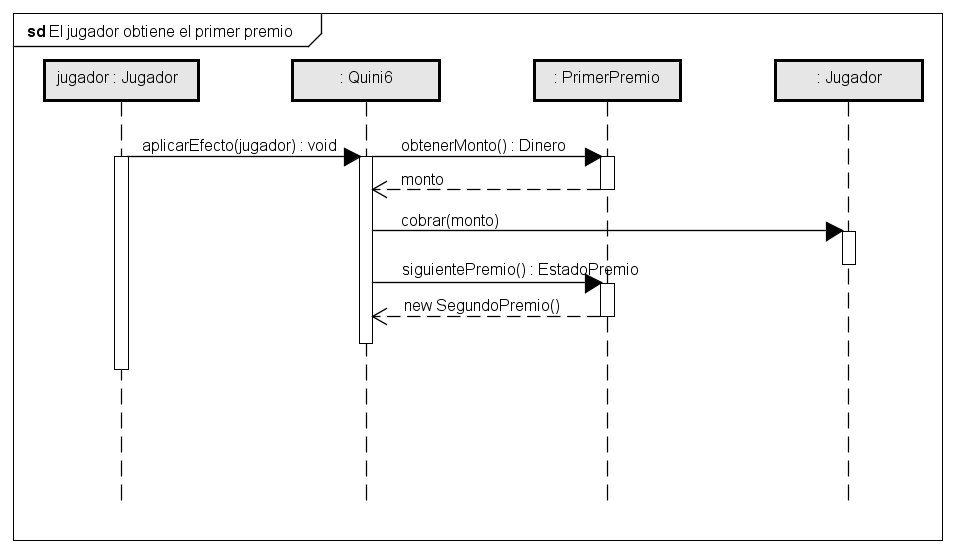
****

Entonces, siguiendo el patrón elegido, se creó una clase *“Premio”* que contiene un *“EstadoPremio”*. El *“EstadoPremio”* es una interfaz que es implementada por las clases concretas *“PrimerPremio”*, *“SegundoPremio”*, *“SinPremio”*.

****

Cada estado de premio es capaz de generar al siguiente premio. Por ejemplo, el primer premio tiene un método concreto que crea al segundo y el segundo al sin premio.

Entonces cada jugador al cobrar el primer premio llama al siguiente premio. El último premio en instanciarse es “SinPremio”. Para esta clase, el método siguiente premio devolverá una referencia a sí mismo.

****