**PATRON STATE**

**Propósito**

Este es un patrón de diseño de comportamiento que permite a un objeto alterar su comportamiento cuando su estado interno cambia. Parece como si el objeto cambiara su clase.

**Dominios de aplicación**

El patrón se utiliza en los siguientes casos:

* El comportamiento de un objeto depende de su estado.
* La implementación de esta dependencia del estado mediante instrucciones condicionales se vuelve muy compleja.

**Participantes**

Son los siguientes:

* Context es una clase concreta que describe los objetos que son máquinas de estados, es decir que poseen un conjunto de estados que pueden ser descritos mediante un diagrama de estados y transiciones. Esta clase mantiene una referencia hacia una instancia de una subclase de la clase State que define el estado en curso.
* State es una clase abstracta que incluye los métodos ligados al estado y que gestionan la asociación con la clase Context.
* ConcreteStateA y ConcreteStateB son subclases concretas que implementan el comportamiento de los métodos relativos a cada estado.

**Estructura**

