## Patrones de Diseño: Patrones de Comportamiento. Tema 5-9: State

#### Descripción del patrón

#### Nombre:

- Estado
- También conocido como Objects for States (objetos para estados)

#### Propiedades:

- Tipo: comportamiento
- Nivel: objeto, componente

#### Objetivo o Propósito:

 Permitir que un objeto se comporte de distinta forma dependiendo de su estado interno, como si cambiase la clase a la que pertenece. Permite cambiar fácilmente el comportamiento de un objeto en tiempo de ejecución.





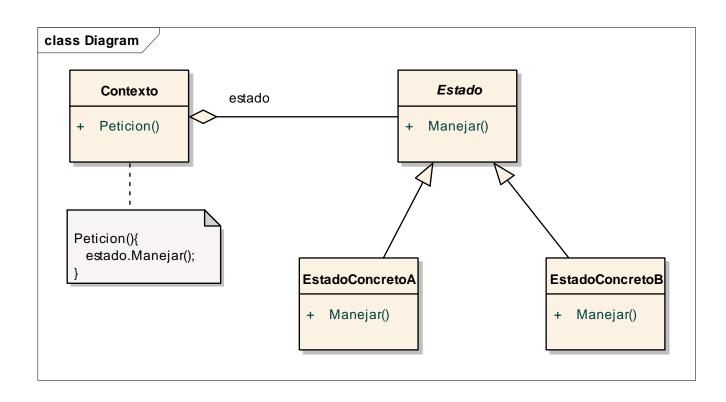
#### Aplicabilidad

- Use el patrón State cuando:
  - El comportamiento de un objeto dependa de su estado y queramos cambiar de comportamiento en tiempo de ejecución.
  - Distintas operaciones de una clase tengan una estructura de control condicional compleja que dependa del estado, permite convertir cada rama en una clase y tratar el estado como un objeto aparte.

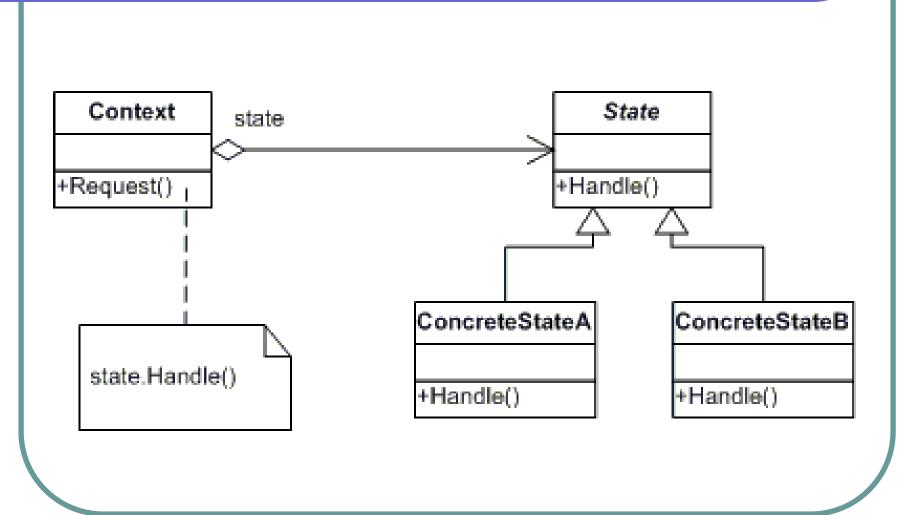




#### Estructura



#### Estructura







#### Estructura. Participantes

- Contexto: Clase que mantiene una referencia al estado actual y es la interfaz que utilizan los clientes.
   Delega todas las llamadas a los métodos específicos del estado en el objeto Estado actual.
- Estado: Interfaz que define los métodos que dependen del estado del objeto.
- EstadoConcreto: Implementa la interfaz Estado.
  Implementa el comportamiento específico de un estado.





#### Consecuencias

- Flexibiliza la creación de nuevos estados y transiciones por lo que se obtiene una visión mucho más clara del comportamiento. Para crear un nuevo estado únicamente hay que crear una nueva clase que implemente la interfaz.
- El flujo de control del código que depende del estado se hace patente en la jerarquía de clases.
- Los estados están ahora encapsulados en objetos por lo que se facilita el reconocimiento de un cambio entre estados.
- Un inconveniente puede ser el gran número de clases asociadas que suele tener este patrón, pero cuando se compara con las largas cadenas de switch/case que se generarían si no se utiliza Estado, puede ser considerado como una ventaja.





#### Patrones relacionados

- Flyweight: Los estados pueden ser compartidos utilizando el patrón flyweight.
- Singleton: La mayoría de los estados son Singletons, especialmente cuando son flyweight.



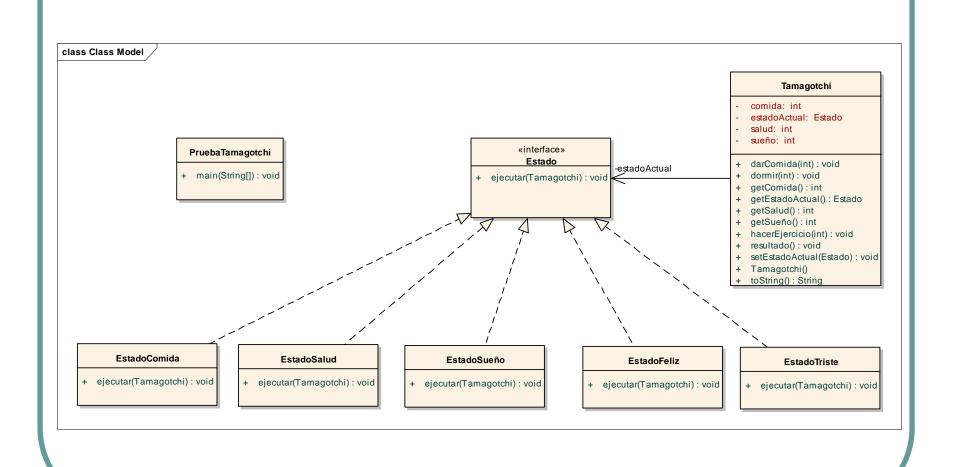


## Código de ejemplo

# Tamagotchi



### Código de ejemplo







## Código de ejemplo

- Identificamos a continuación los elementos del patrón:
  - Contexto: Tamagotchi
  - Estado: Estado
  - EstadoConcreto: EstadoComida, EstadoSalud, EstadoSueño, EstadoFeliz, EstadoTriste.



