

Para la clase "EstadoDispositivo" se utiliza la estrategia de mapeo SINGLE TABLE ya que sus clases hijas "AhorroDeEnergia", "Prendido" y "Apagado" no tienen atributos y solo aportan comportamiento.

Para la clase "Dispositivo" se utiliza la estrategia de mapeo JOINED ya que sus clases hijas "Inteligente" y "Standard" poseen atributos similares pero también muchos atributos distintos, por esto también no convendría usar single table, ya que estos atributos distintos serían nulos en la tabla única.

Para la clase Inteligente y su clase hija "ModuloAdaptador" no se utiliza ninguna estrategia de mapeo, ya que lo consideramos como sobrediseñar, en vez de una estrategia simplemente se agregó un atributo booleano ("estaAdaptado") a la clase "Standard".

Para la clase "ActuadorBase" se utiliza la estrategia de mapeo SINGLE TABLE ya que sus clases hijas "Apagar", "BajarIntensidad", "Prender", "PasarAModoAhorroEnergia", "SubirIntensidad" y "SetearTimer" solo tienen el atributo DisplInteligente y solo aportan comportamiento.

Se eliminó la herencia de la clase Polygon por parte de "ZonaGeografica", ya que ZonaGeografica solo heredaba comportamiento, ahora pasa a ser una composición.(polygon pasa a ser solo un atributo).

Para la clase "Ubicable" se utiliza la estrategia de mapeo JOINED ya que sus clases hijas "ZonaGeografica", "Usuario" y "Transformador" poseen atributos distintos pero no conviene utilizar "table per class", porque la clase padre, al tener una colección, complejiza la aplicación de esa estrategia.

Para la clase "Criterio" se utiliza la estrategia de mapeo SINGLE TABLE ya que sus clases hijas "RangoExcluyente", "Menor", "Mayor", "Igualdad" y "Distinto" solo aportan comportamiento.

Para la clase "Usuario" se utiliza la estrategia de mapeo JOINED ya que sus clases hijas "Administrador" y "ClienteResidencial" poseen atributos similares pero también muchos atributos distintos.



