Patrones de Diseño: Patrones Estructurales. Tema 4-7: Flyweight

Descripción del patrón

- Nombre:
 - Peso ligero
- Propiedades:
 - Tipo: estructural
 - Nivel: objeto, componente
- Objetivo o Propósito:
 - Reducir el número de objetos detallados de muy bajo nivel en un sistema mediante la compartición de objetos.





Definiciones

- Definiciones a tener en cuenta:
 - Peso ligero: Objeto compartido que puede usarse a la vez en varios contextos.
 - Estado intrínseco: Se guarda con el propio objeto, consiste en información que es independiente de su contexto y que puede ser compartida.
 - Estado extrínseco: Depende del contexto y cambia con él, no puede ser compartido.
- Los pesos ligeros modelan conceptos o entidades que normalmente son demasiado numerosos como para ser representados como objetos.
- <u>Ejemplo:</u> Un editor de documentos tendrá un peso ligero para cada letra del alfabeto con su código de carácter (estado intrínseco), sin embargo las coordenadas de posición en el documento y su estilo se obtienen con los algoritmos de maquetación y formateado (estado extrínseco).
- En el API de Java se utiliza el patrón para representar bordes (BorderFactory).





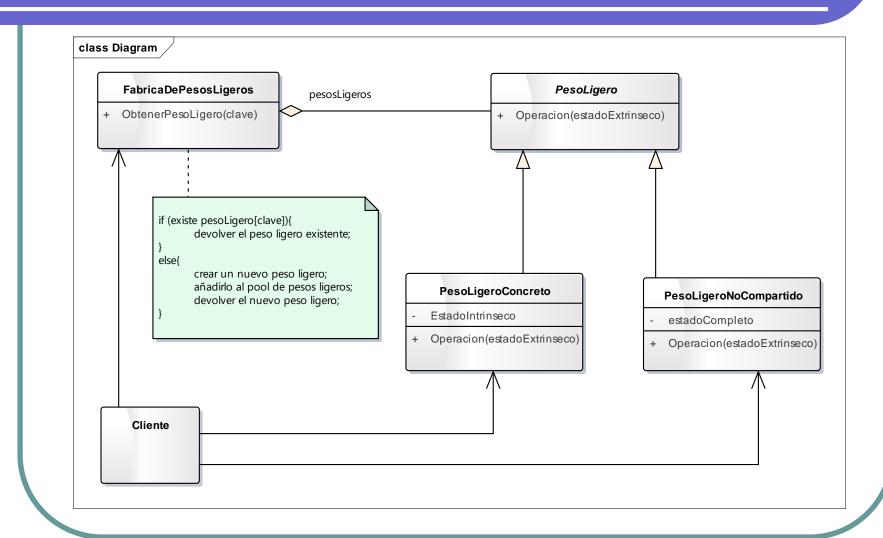
Aplicabilidad

- Use el patrón Flyweight cuando se cumplan todas estas condiciones:
 - Una aplicación utiliza muchos objetos con características similares.
 - Los costes de almacenamiento de tantos objetos son elevados.
 - Las características de los objetos se pueden dividir en similares (estado intrínseco) y diferentes (estado extrínseco), permitiendo compartir las comunes.
 - Los grupos de objetos similares pueden reemplazarse por un objeto compartido cuyas partes diferentes han sido eliminadas.
 - La aplicación no depende de la identidad de un objeto.
 Puesto que los objetos peso ligero pueden ser compartidos.





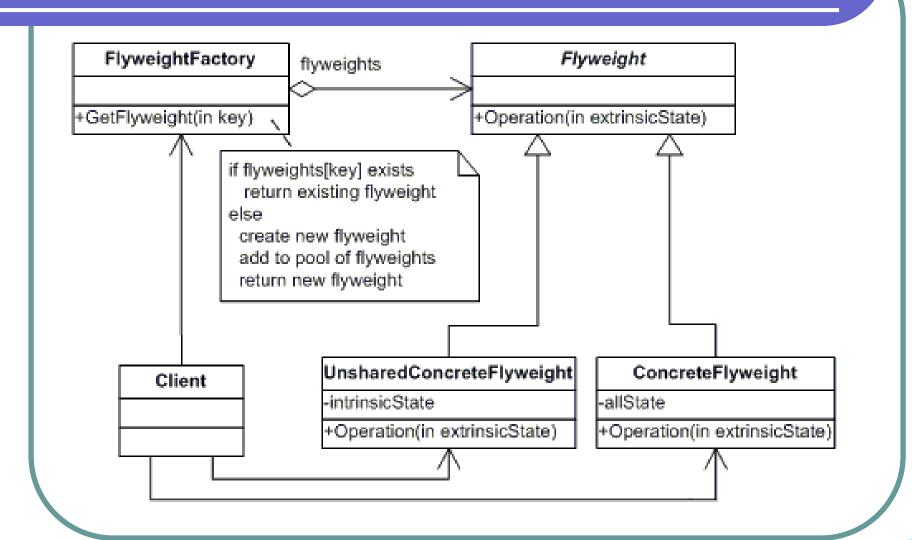
Estructura







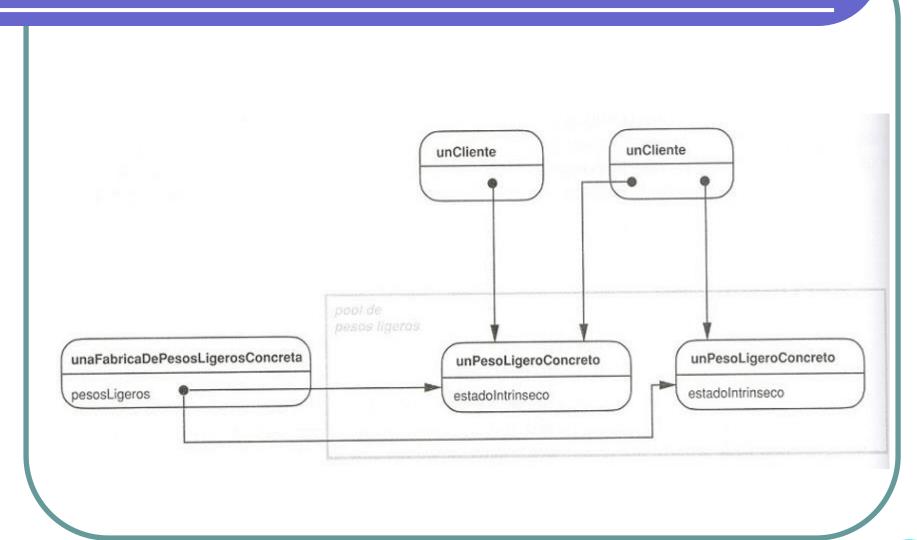
Estructura







Estructura. Comportamiento







Estructura. Participantes

- **PesoLigero:** Declara una interfaz para recibir un estado extrínseco y actuar sobre el.
- PesoLigeroConcreto: Implementa la interfaz PesoLigero y permite almacenar el estado intrínseco, si lo hay. Un objeto de este tipo puede ser compartido, por lo que debe ser independiente del contexto.
- **PesoLigeroConcretoNoCompartido:** Implementa la interfaz PesoLigero pero sus instancias no necesitan ser compartidas.
- FabricaDePesosLigeros: Encargada de crear y controlar pesos ligeros. Cuando un cliente pide un peso ligero la fábrica le proporciona una instancia concreta ya existente o crea uno nuevo.
- Cliente: Mantiene una referencia a los pesos ligeros. Calcula o guarda el estado extrínseco de los pesos ligeros.





Consecuencias

- Se reduce el número de objetos que hay que gestionar.
- Se pueden incrementar los costes en tiempo de ejecución si se realizan transferencias, búsquedas o cálculos de su estado extrínseco.
- Cuanto mayor sea el número de objetos peso ligero compartidos, mayor ahorro de espacio tendremos.





Patrones relacionados

- Abstract Factory: Utilizado para proporcionar acceso a los objetos Flyweight asegurando su correcta compartición.
- Composite: Se suele utilizar para proporcionar una estructura jerárquica con nodos hojas compartidos.
- State y Strategy: Se utiliza Flyweight para implementar estos patrones.





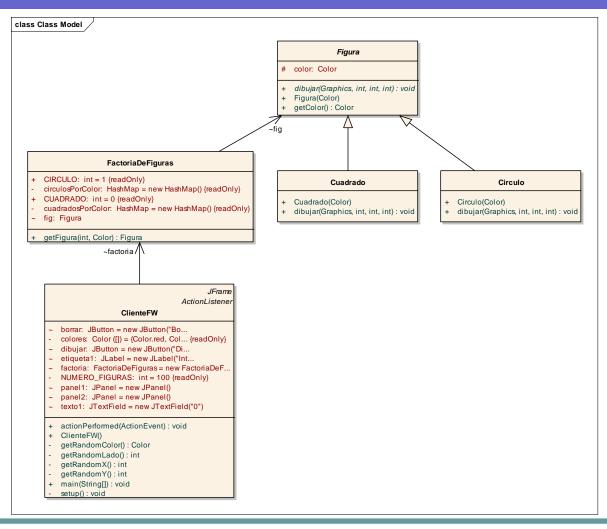
Código de ejemplo

HURAS





Código de ejemplo







Código de ejemplo

- Identificamos a continuación los elementos del patrón:
 - PesoLigero: Figura.
 - PesoLigeroConcreto: Cuadrado, Circulo.
 - FabricaDePesosLigeros: FactoriaDeFiguras.
 - Cliente: ClienteFW.



