

Introducción a los patrones GOF

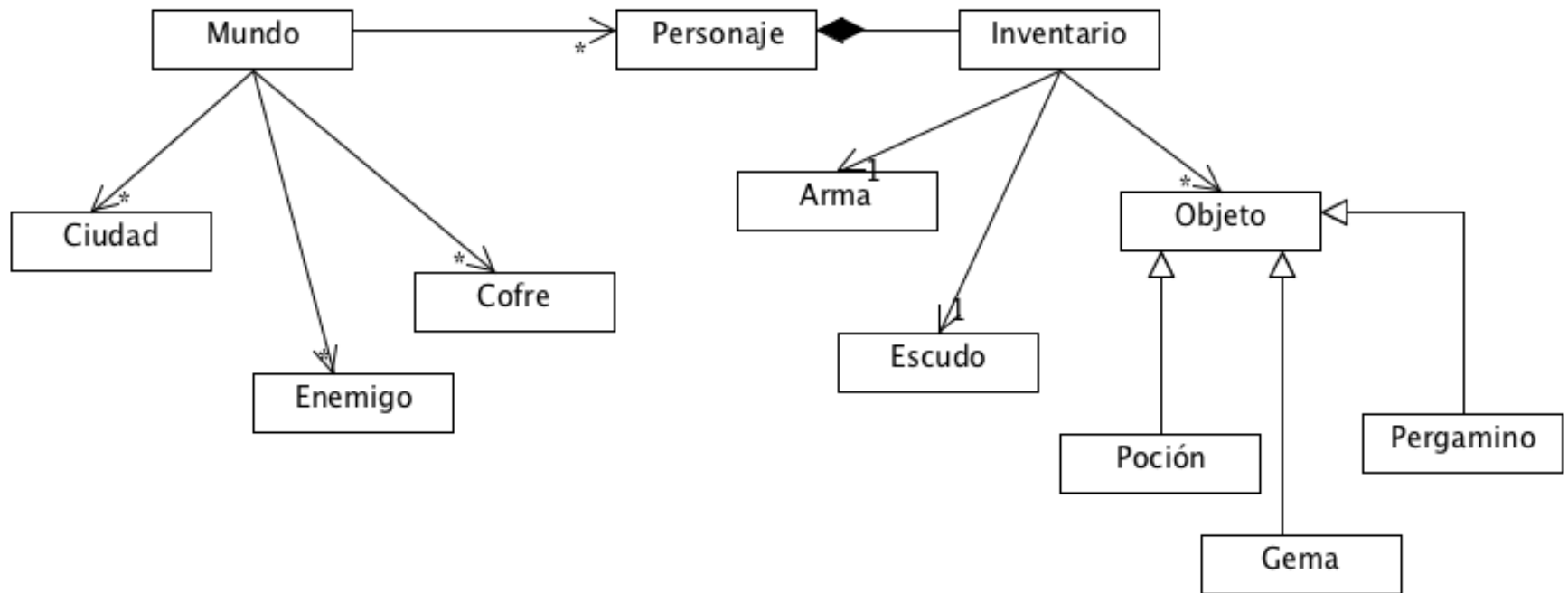
Diseño de Sistemas Software

Patrones GOF

- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
Design Patterns: Elements of Reusable Object-Oriented Software
Addison-Wesley, 1994
- 3 tipos
 - **Creacionales**: abstract factory, builder, factory method, prototype, singleton.
 - **Estructurales**: adapter, bridge, composite, decorator, facade, flyweight, proxy.
 - **De comportamiento**: chain of responsibility, command, interpreter, iterator, mediator, memento, observer, state, strategy, template method, visitor.

- Diseña un diagrama de clases para un juego de rol con las siguientes características:
 - Hay numerosos personajes que se mueven por un mundo abierto, en el que pueden encontrar distintas ciudades.
 - Los personajes tienen un inventario en el que guardan un arma, un escudo y los objetos que van consiguiendo.
 - Se pueden conseguir objetos dentro de cofres o al derrotar a los pequeños enemigos que merodean por el mundo.
 - Los objetos pueden ser pociones, gemas, pergaminos, etc.

Solución

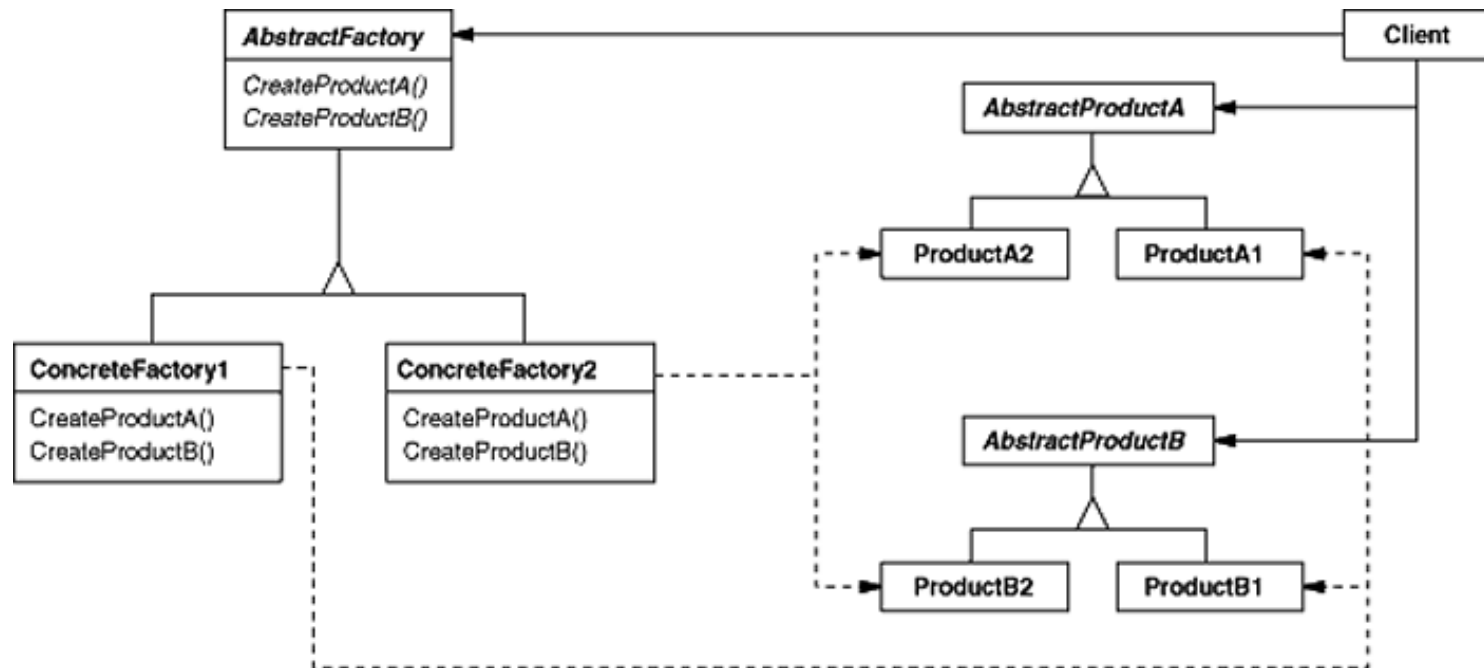


Aplicación de un patrón creacional

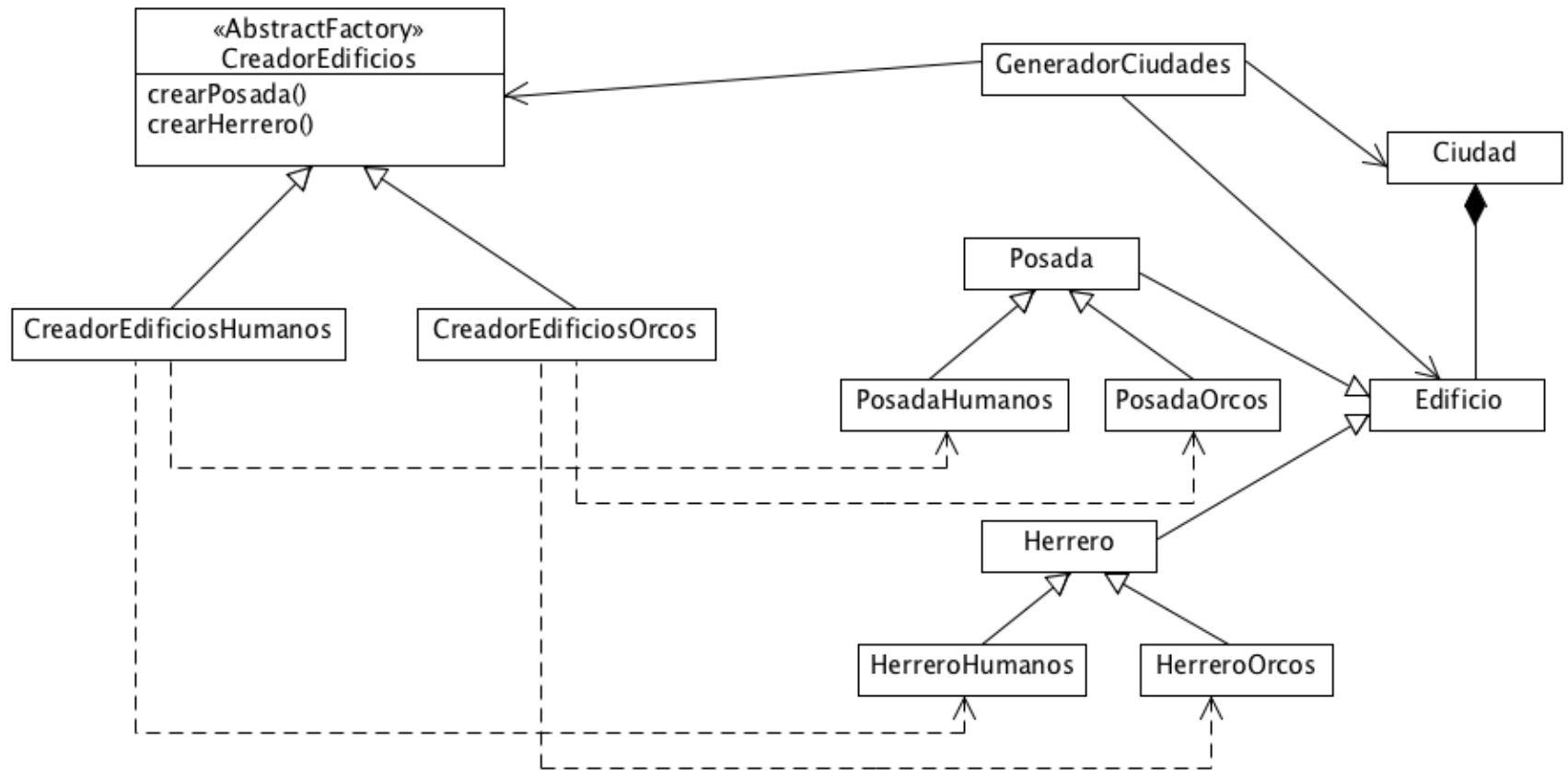
- Ampliación del juego
 - Queremos que las ciudades se generen automáticamente a medida que los personajes recorren el mundo abierto. La clase *GeneradorCiudad* se encargará de crearlas.
 - Las ciudades están compuestas de distintos tipos de edificios: posadas, herreros, tiendas, etc.
 - Los edificios son distintos en función de la raza a la que pertenecen. En el juego existen dos razas distintas, humanos y orcos, aunque se contempla que podrá haber más en el futuro.
 - En ningún caso se podrán mezclar edificios de razas distintas en una misma ciudad.

Abstract factory (creacional)

- **Abstract Factory:** provee una interfaz para crear familias de objetos producto relacionados o que dependen entre sí, sin especificar sus clases concretas.



Solución

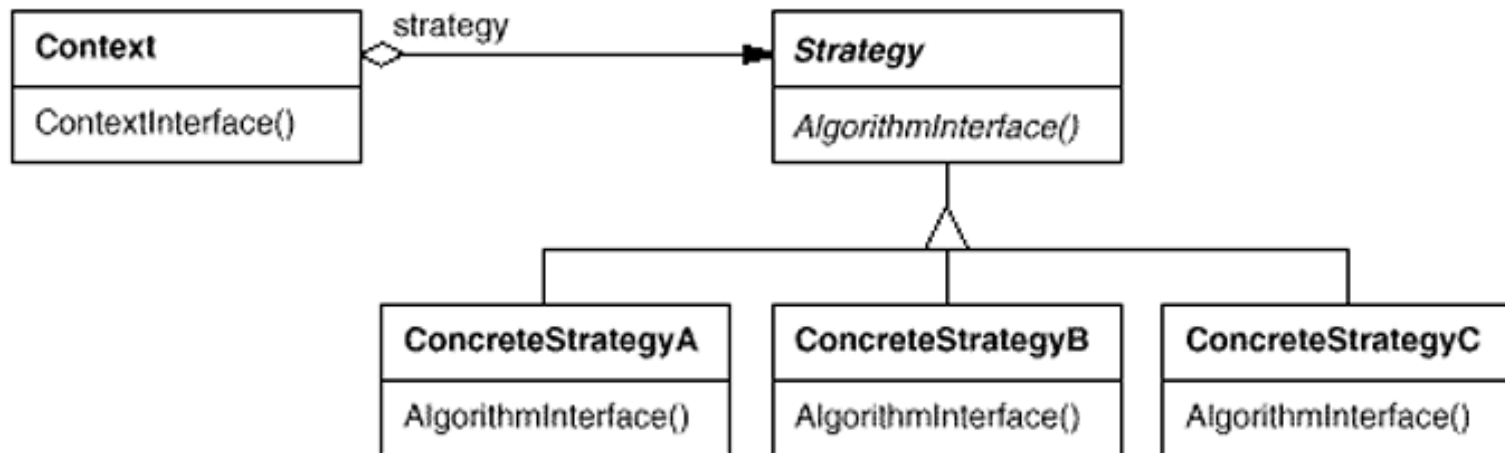


Aplicación de un patrón de comportamiento

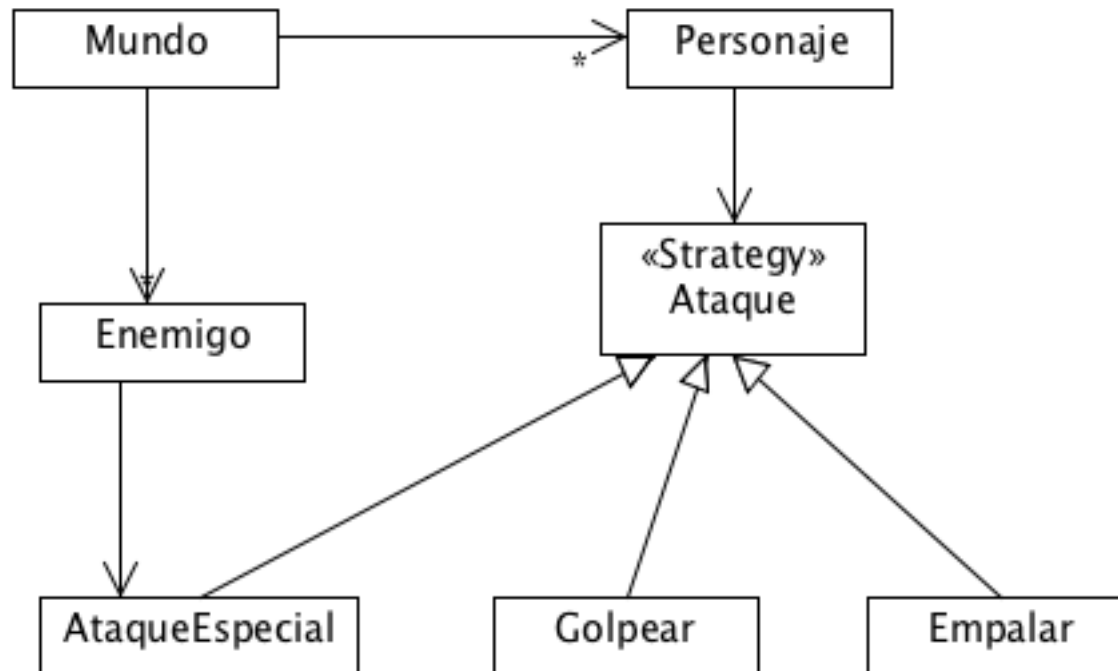
- Ampliación del juego
 - En el juego los personajes podrán usar distintas armas, que se diferencian en los puntos de ataque (fuerza) que tienen.
 - En cada ataque, los personajes pueden elegir entre varios tipos de ataque: golpear, empalar o ataques especiales; ya que cada ataque afecta de forma distinta a cada enemigo.
 - Los personajes sólo podrán emplear los ataques especiales cuando los hayan obtenido al derrotar enemigos.

Strategy (de comportamiento)

- **Strategy:** define una jerarquía de clases que representan algoritmos, los cuales son intercambiables. Estos algoritmos pueden ser intercambiados por la aplicación en tiempo de ejecución.



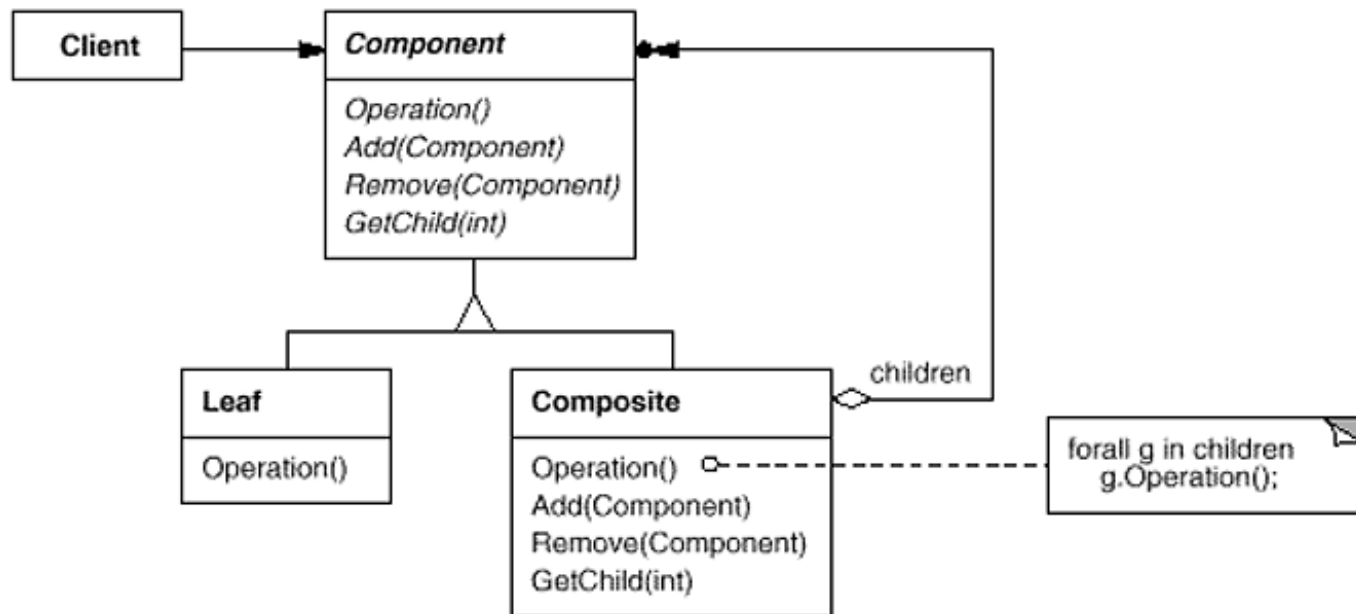
Solución



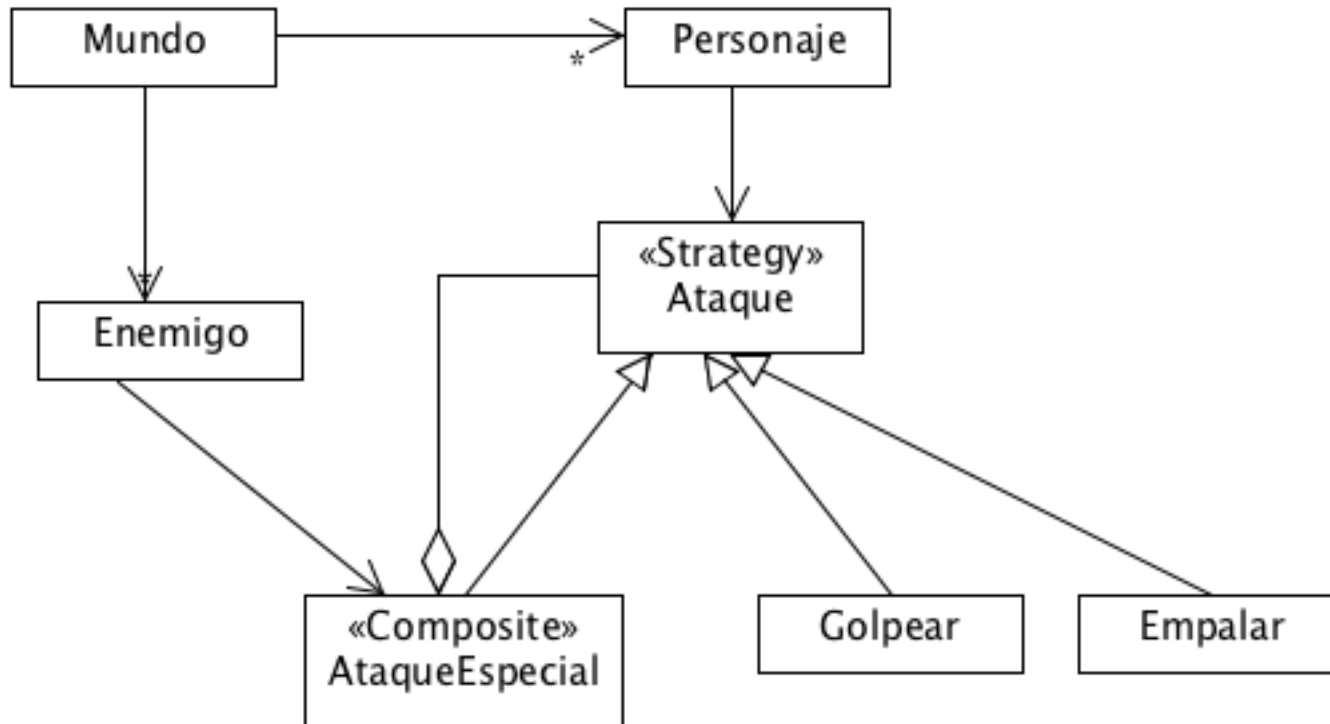
- Ampliación del juego
 - Algunos ataques especiales consisten en combos: una combinación de otros ataques que el personaje ejecutará en secuencia

Composite (estructural)

- **Composite:** compone objetos en estructuras de árboles para representar jerarquías parte-todo. Permite que los clientes traten de manera uniforme a los objetos individuales y a los complejos.



Solución



- Para la próxima sesión, estudiar los patrones que se publicarán en Moodle.