# Patrones de Diseño: Patrones de Comportamiento. Tema 5-5. Iterator

## Descripción del patrón

#### Nombre:

- Iterador
- También conocido como cursor

#### Propiedades:

- Tipo: comportamiento
- Nivel: objeto, componente

#### Objetivo o Propósito:

 Proporcionar una forma coherente de acceder secuencialmente a los datos de una colección, independientemente del tipo de colección.





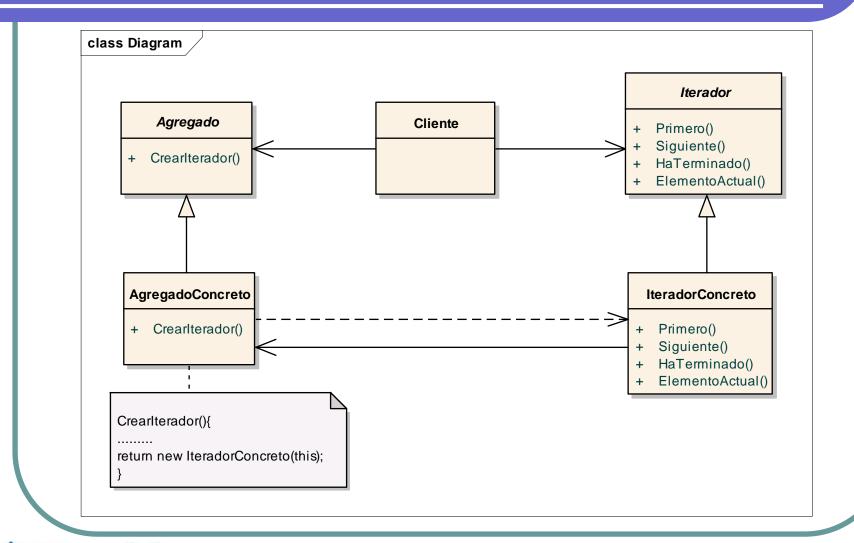
#### Aplicabilidad

- Use el patrón Iterador cuando:
  - Queramos proporcionar una forma uniforme, coherente e independiente de la implementación, de desplazarse por los elementos de una colección.
  - Queramos permitir el recorrido de múltiples colecciones, permitiendo que distintos clientes naveguen simultáneamente por la misma colección.
- En el framework de colecciones de Java se utiliza el patrón *Iterator* mediante la interface *java.util.Iterator* que proporciona la capacidad de iterar sobre cualquier objeto de negocio. Define los métodos básicos de navegación *hasNext* y *next*.



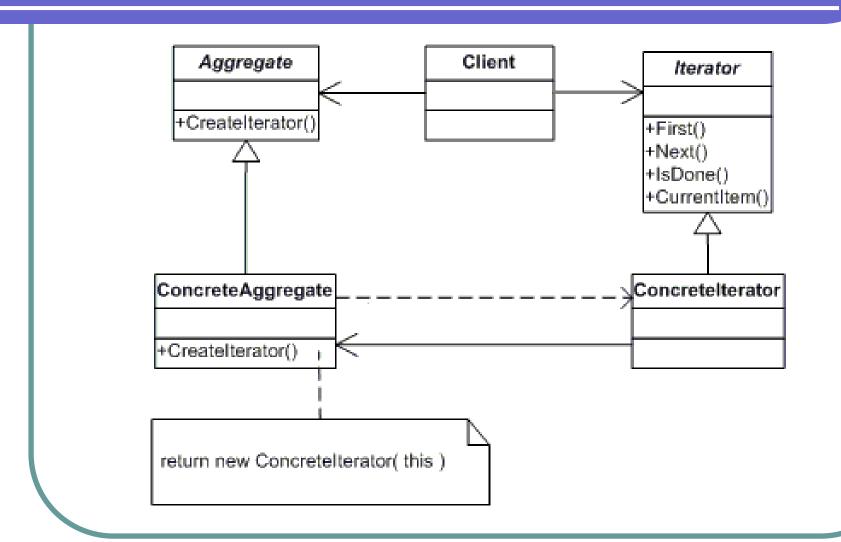


#### Estructura





#### Estructura







#### Estructura. Participantes

- Iterador: Interfaz en la que se definen los métodos de iteración.
  Como mínimo, se definen los métodos primero, siguiente, hayMas y recuperación del elemento actual.
- IteradorConcreto: Clases "iteradoras" que implementan la interfaz Iterador. Sus instancias se crean por medio de AgregadoConcreto. Debido al alto acoplamiento con esta clase, a menudo IteradorConcreto suele ser una clase interna de AgregadoConcreto.
  - Mantiene la posición actual en el recorrido del agregado.
- Agregado: Interfaz que define el método de fabricación para crear un iterador.
- AgregadoConcreto: Implementa la interfaz Agregado y crea los iteradores concretos según se necesiten.





#### Consecuencias

- Podemos recorrer una estructura de objetos sin conocer los detalles internos de sus clases.
- Al definir una interfaz uniforme para los iteradores concretos, se simplifica el recorrido de las colecciones y se permite utilizar polimorfismo.
- Al tener que recorrer agregados complejos podemos hacerlo de muchas formas distintas. Facilitan cambiar el algoritmo de recorrido.
- Los iteradores permiten a los clientes manejar múltiples puntos de navegación para la misma colección, ya que un iterador es como un puntero de una colección. Simplemente instanciando nuevos iteradores, podemos conseguir distintos puntos de navegación en una misma colección.





#### Patrones relacionados

- Factory Method: Se puede emplear un Factory Method para determinar que Iterator instanciar. Los iteradores polimórficos se basan en este patrón para instanciar las subclases apropiadas de Iterador.
- Composite: Los iteradores suelen aplicarse a estructuras recursivas como los compuestos.
- Memento: Suelen usarse conjuntamente. Un iterador puede usar un memento para representar el estado de una iteración. El iterador almacena el memento internamente.





# Código de ejemplo

# Universitatios



## Código de ejemplo

