

A thick black L-shaped frame is positioned on the left and bottom edges of the slide, framing the central text.

BUILDER PATRÓN DE DISEÑO

Diseño y Arquitectura de Software

Nombre y clasificación

Builder

Se clasifica dentro de los patrones de creación

Son los que facilitan la creación de nuevos objetos, de tal forma que el proceso de creación pueda ser desacoplado de la implementación del resto del sistema.

Ocultan como las implementaciones concretas necesitan ser creadas y como se combinan entre sí

Otros nombres:

BUILDER- CONSTRUCTOR

Intención

- Separa la creación de un objeto complejo de su estructura , de tal forma que el mismo proceso de construcción nos puede servir para crear representaciones diferentes.
- Abstrae el proceso de creación de un objeto complejo centralizando dicho proceso en un único punto, de tal forma que el mismo proceso de construcción pueda crear representaciones diferentes.

Motivación

- Este es un patrón bastante simple pero muy útil, nos permite crear objetos complejos a través de uno mas simple.
- Es común encontrarnos con situaciones en las cuáles tenemos que crear objetos compuestos de forma manual y repetidas veces, lo que nos lleva a tener que establecer cada propiedad del objeto si ésta además tiene objetos compuestos dentro, tenemos que crearlos primero para después ser asignados al objeto que estamos creando

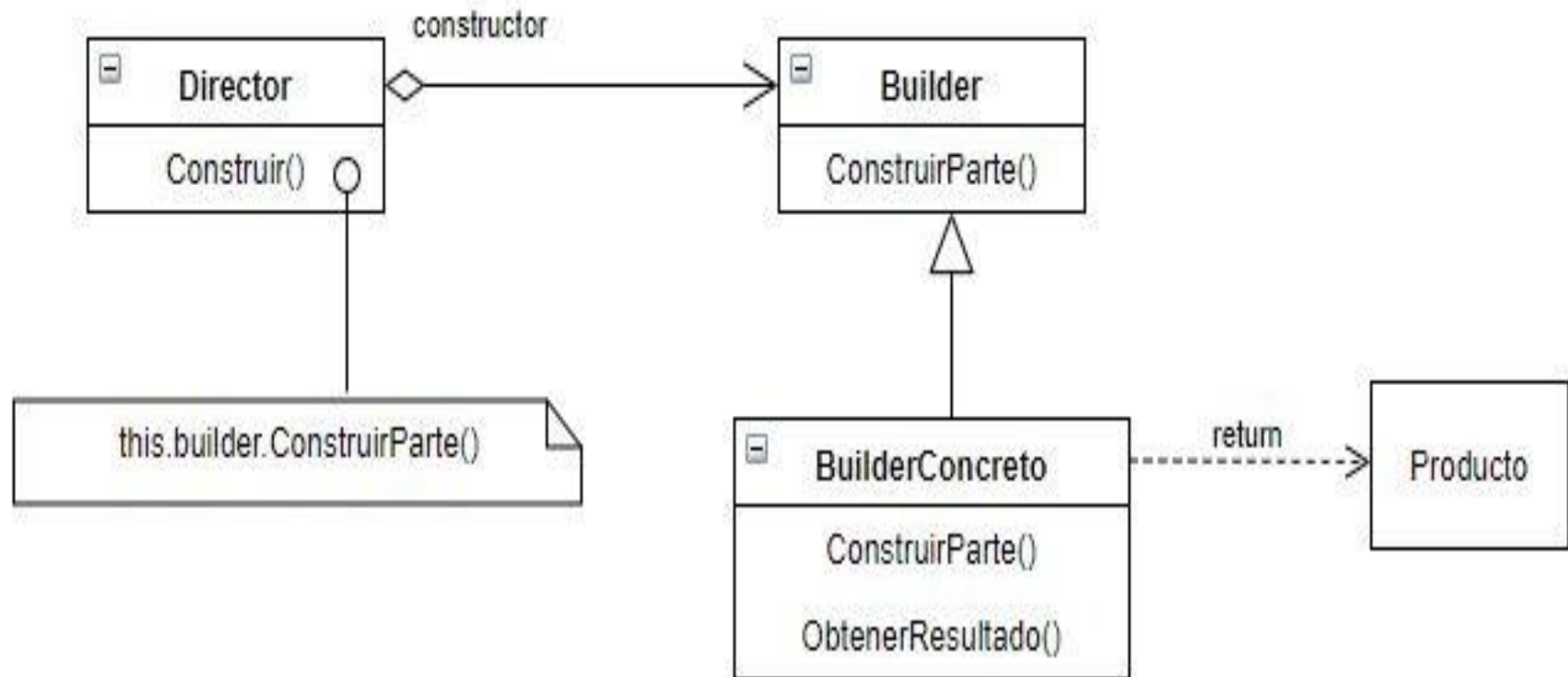
Aplicación

Utilizar cuando...

El algoritmo para crear un objeto complejo deba ser independiente de las partes que componen el objeto y de como se ensamblan.

El proceso de construcción debe permitir distintas representaciones para el objeto que es construido.

Estructura:



Participantes

- Builder: especifica una interfaz abstracta para crear un objeto complejo producto.
- ConcreteBuilder: construye y ensambla las partes del producto implementando interfaz del constructor.
- Director: obtiene las partes que forman el producto y lo construye usando la interfaz del constructor.
- Product: Representa el objeto complejo en construcción.

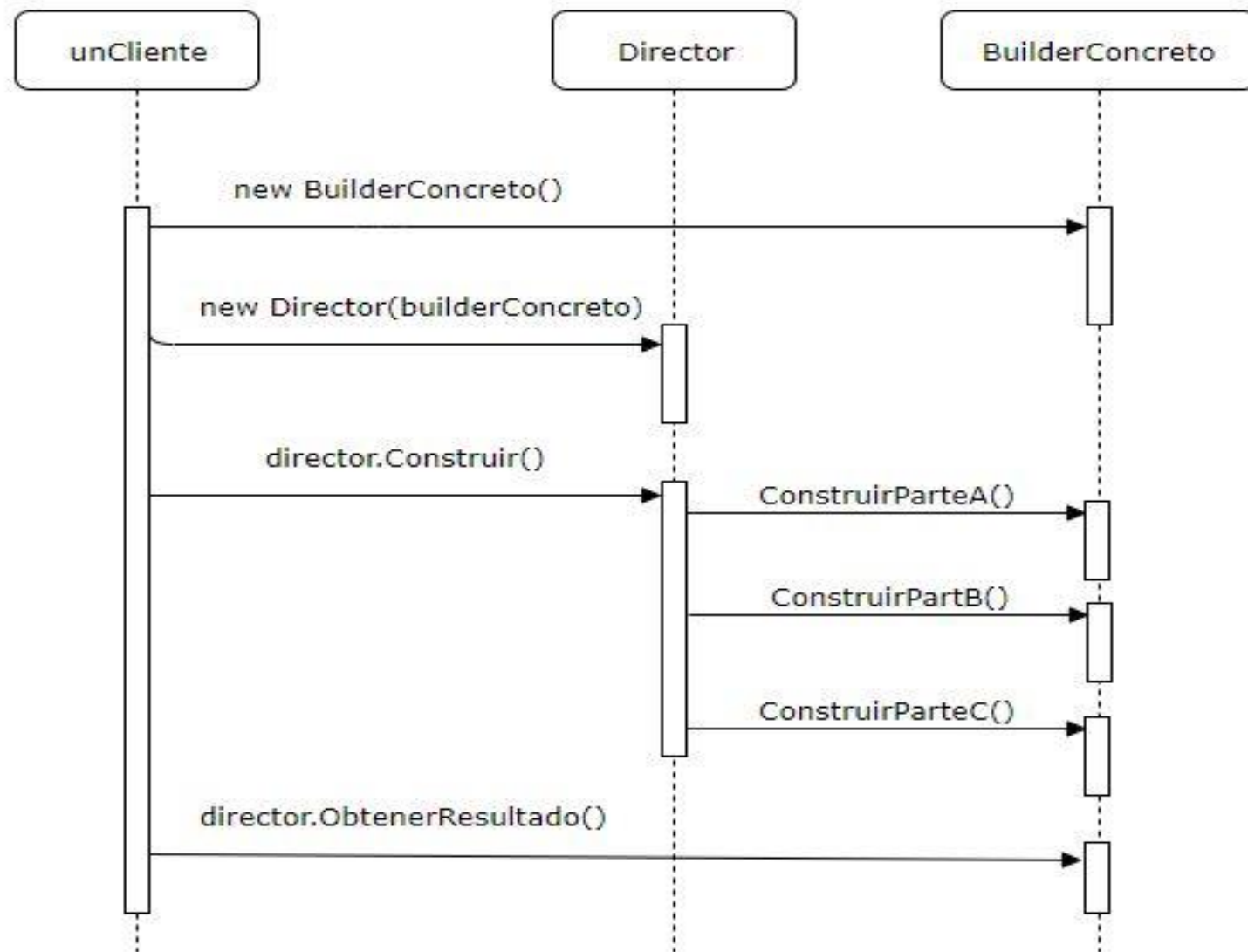
Colaboraciones:

El cliente crea el objeto director y lo configura con el objeto constructor deseado.

El director notifica al builder cada vez que hay que construir una parte del producto.

El constructor gestiona las peticiones del director y añade partes al producto.

El cliente obtiene el producto del constructor.



Consecuencias

Desacopla al director del constructor, permitiendo nuevas representaciones de objetos cambiando el constructor

Aísla a los clientes de la estructura interna del producto

Permite controlar la construcción paso a paso de un objeto complejo

Se nos permite variar la representación interna del producto; como el producto se construye a través de una interfaz abstracta, para cambiar la representación interna del producto basta con definir un nuevo tipo de constructor.

Se nos permite separar el código de la construcción y la representación.
Nos da un control muy fino sobre el proceso de construcción de un producto.

EJEMPLOS DE CODIGO



USOS CONOCIDOS:

En .net Framework hay pocos usos conocidos SqlConnectionStringBuilder se usa para poder añadir partes de la cadena de conexión por separado.

UriBuider: parecido al anterior , se usa para crear URL´s indicando partes de la misma , puerto, host, etc.

IwebHostBuilder: a través de builder se puede obtener el IwebHost configurado adecuadamente y que puede ser ejecutado.

Patrones relacionados

Los patrones de Abstract Factory y Builder son muy similares , ya que los dos pueden realizar la construcción de objetos complejos .

La principal diferencia entre ellos es que el patrón Builder se centra en construir el objeto complejo paso a paso, mientras que el patrón Abstract Factory da mas importancia ala similitud de los objetos.