



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

CLASE 5

PATRONES DE DISEÑO

Análisis y Diseño de Sistemas 2

Aux. Kenny Miguel Eguizabal

Patrones de Diseño

- En Ingeniería de Software, un patrón de diseño es una solución general reutilizable a un problema que ocurre comúnmente en un contexto determinado en el diseño de software.
- Un patrón de diseño no es un diseño o solución final que se puede transformar directamente en código fuente. Es una descripción o una plantilla para la forma de resolver un problema que se puede utilizar en muchas situaciones diferentes.

Patrones de Diseño

- Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software. Se formalizaron como mejores prácticas que el programador puede utilizar para resolver los problemas comunes en el diseño de una aplicación o sistema.

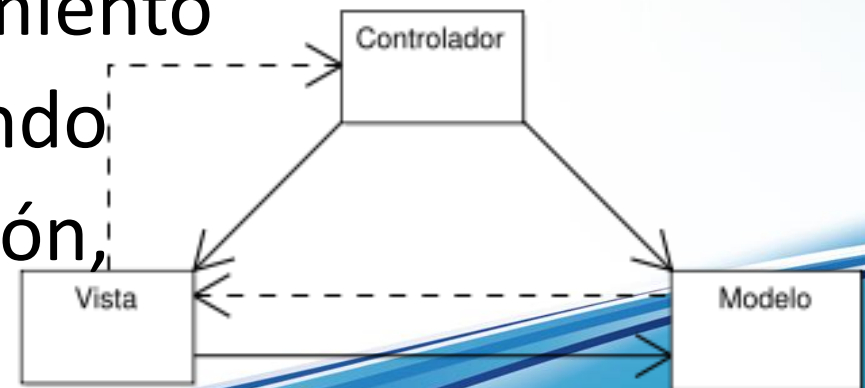


Patrones de Diseño

Los patrones de diseño se agruparon inicialmente en las categorías:

- Patrones de Creación (*Creacionales*)
- Patrones de Estructura (*Estructurales*)
- Patrones de Comportamiento

Estos se describen utilizando los conceptos de delegación, agregación y la consulta.

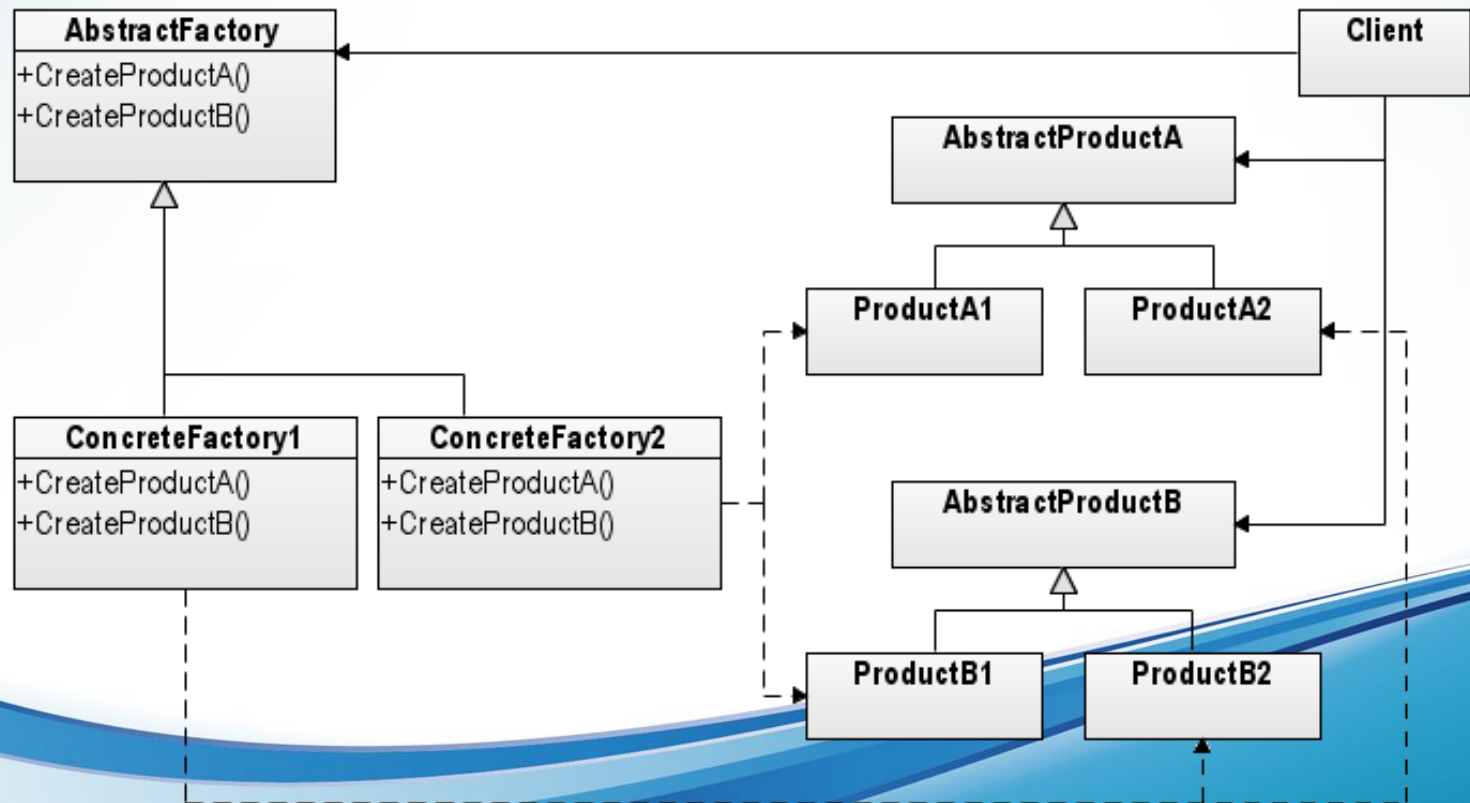


Patrones de Creación

- Los patrones creacionales, son los que crean objetos y aíslan sus detalles, en lugar que tener que crear instancias directamente objetos. Esto le da a su programa más flexibilidad para decidir que objetos se deben crear para un caso específico, de forma que su código no dependa de los tipos de objeto que hay, y por lo tanto no tenga que cambiarlo cuando añada un nuevo tipo de objeto.

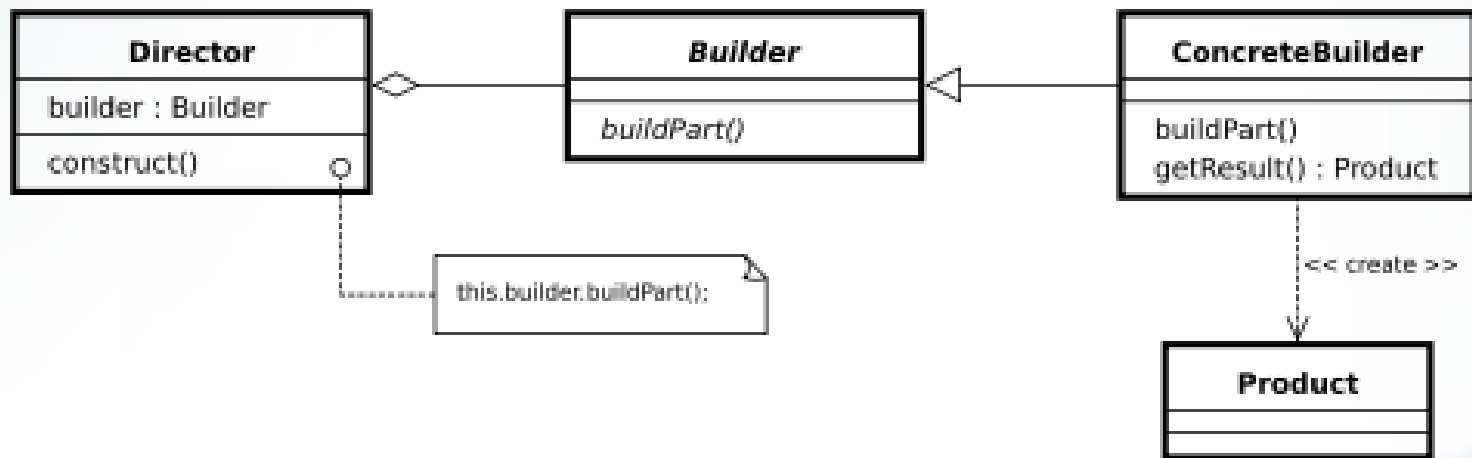
Patrones de Creación

- **Abstract Factory:** Grupo de objetos que tienen un tema en común



Patrones de Creación

- **Builder:** Construye objetos complejos separando su construcción y representación.



Patrones de Creación

- **Singleton:** Restringe la creación de objetos de una clase a una sola instancia.

Singleton
<u>- singleton : Singleton</u>
- Singleton() <u>+ getInstance() : Singleton</u>

Singleton

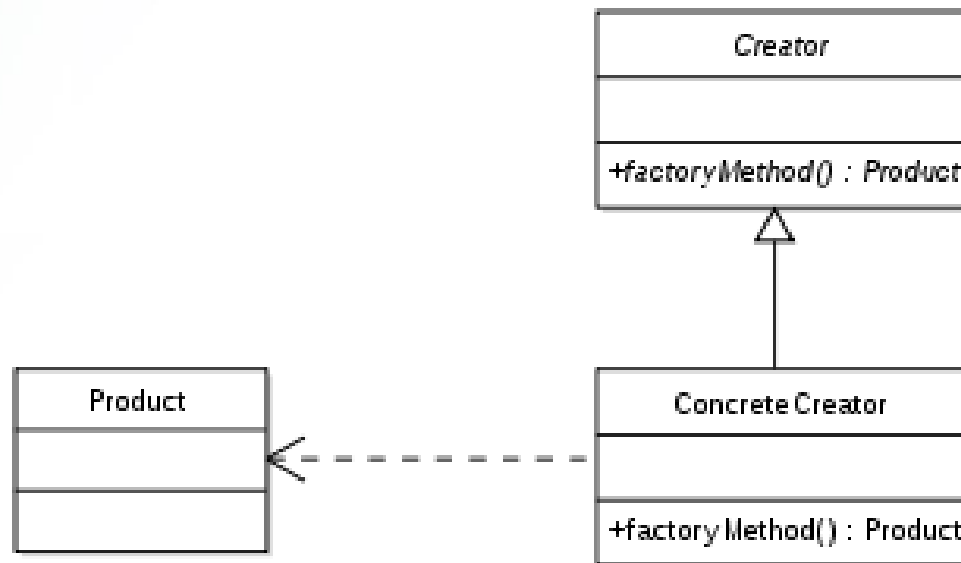
- El patrón singleton es un patrón de diseño que restringe la creación de instancias de una clase a un solo objeto. Esto es útil cuando se necesita exactamente de un objeto para coordinar las acciones de todo el sistema.
- El concepto es a veces generalizado a sistemas que operan de manera más eficiente cuando sólo existe un objeto, o que restringe la creación de instancias a un cierto número de objetos. El término viene del concepto matemático de un singleton.

Singleton

Hay críticas al uso del patrón singleton, ya que algunos consideran que es un anti-patrón, pues se considera que su uso excesivo, introduce restricciones innecesarias en situaciones en las que en realidad no se requiere una única instancia de una clase, y se introduce estados globales en una aplicación.

Patrones de Creación

- **Factory Method:** Crea objetos sin especificar exactamente la clase de crear.



Factory Method

- Este patrón utiliza métodos de fábrica para hacer frente al problema de la creación de objetos sin especificar la clase exacta de objeto que se va a crear.
- Esto se hace mediante la creación de objetos a través de un método de fábrica, que o bien se especifica en una interfaz (clase abstracta) e implementado en las clases (clases concretas) en la ejecución; o bien se implementa en una clase base (opcionalmente como método de plantilla). Que puede ser anulada cuando se hereda en las clases derivadas y no por un constructor.

¡GRACIAS!