Patrones de Diseño: Patrones de Creación. Tema 3-1: Introducción

Patrones de Creación

- Los patrones de creación muestran la guía de cómo crear objetos cuando sus creaciones requieren tomar decisiones. Estas decisiones normalmente serán resueltas dinámicamente decidiendo que clases instanciar o sobre que objetos se delegarán responsabilidades.
- La valía de los patrones de creación nos dice como estructurar y encapsular estas decisiones. A menudo hay varios patrones de creación que se pueden aplicar en una situación o incluso combinarlos. En otros casos se debe elegir tan solo uno de ellos.





Patrones de Creación

- Tienen las siguientes capacidades:
 - Instanciación genérica: Permite crear objetos en un sistema sin tener que identificar una clase específica en el código.
 - **Simplicidad:** Algunos de los patrones facilitan la creación de objetos, por lo que no se tendrá que escribir gran cantidad de código para crear un objeto.
 - Restricciones de creación: Algunos patrones fuerzan restricciones en el tipo o el número de objetos que pueden ser creados en el sistema.





Clasificación Patrones de Creación

- Abstract Factory: Proporciona un contrato para la creación de familias de objetos relacionados o dependientes sin tener que especificar su clase concreta.
- Builder: Simplifica la creación de objetos complejos definiendo una clase cuyo propósito es construir instancias de otra clase.
- Factory Method: Permite definir un método estándar para crear un objeto, además del constructor propio de la clase, si bien la decisión del objeto a crear se delega en las subclases.





Clasificación Patrones de Creación

- Prototype: Facilita la creación dinámica al definir clases cuyos objetos pueden crear copias de sí mismos.
- **Singleton:** Permite tener una única instancia de una clase en el sistema, además de permitir que todas las clases tengan acceso a esa instancia.



Resumen Patrones de Creación

- Hay dos formas de parametrizar un sistema: uno es con la herencia y otro es con la composición. Ejemplo del primer método es el Factory Method y ejemplo del segundo son los patrones Abstract Factory, Builder y Prototype.
- Abstract Factory produce objetos de varias clases.
- Builder construye un producto complejo incrementalmente.
- Prototype hace que su objeto fábrica construya un producto copiando un objeto prototípico.
- Muchas veces los diseños comienzan usando el patrón
 Factory Method y luego evolucionan hacia los otros patrones de creación.



