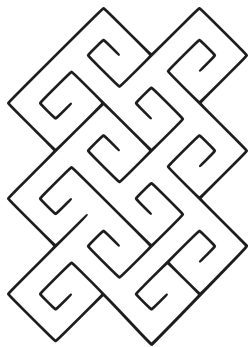


# Patrones de diseño



# Agenda

- Definiciones
- Clasificación
- Ejercicios



# Agenda

- Definiciones
- Clasificación
- Ejercicios



# Definición

"Un patrón describe un problema que ocurre una y otra vez, y luego describe el núcleo de la solución a dicho problema, de forma tal que se puede usar esta solución un millón de veces, sin repetir jamás la forma de aplicarla"

Christopher Alexander

"Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice."

Christopher Alexander

O más simplemente...

"Un patrón es una solución a un problema, dado un contexto."

GOF

# ¿Por qué patrones?

- Conocidos
- Convencionales
- Documentados
- Simples
- Comprobados

Facilitan comunicación

Fáciles de detectar

Describe el contexto

Reducen complejidad

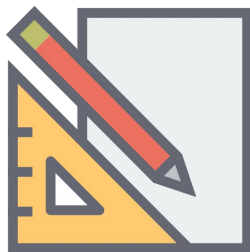
Se sabe que funcionan

# Agenda

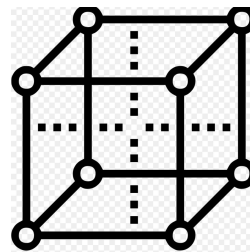
- Definiciones
- Clasificación
- Ejercicios



# Clasificación de patrones



**Patrones creacionales**



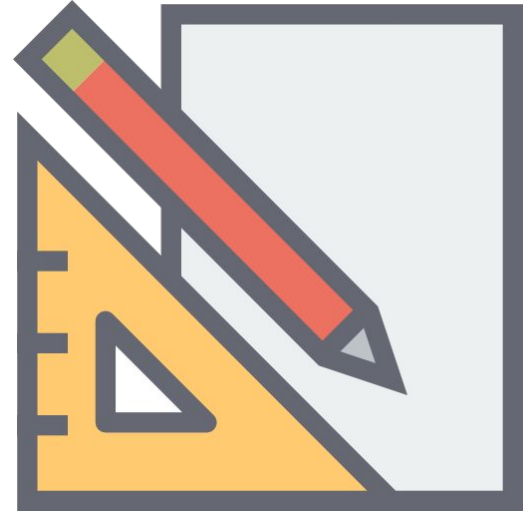
**Patrones estructurales**



**Patrones de comportamiento**

# Creacionales

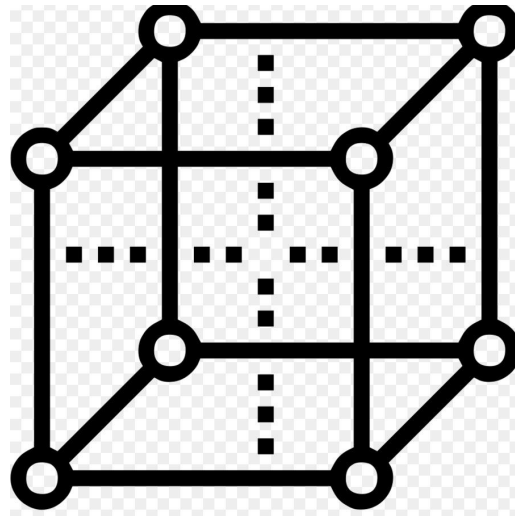
- Abstract Factory
- Factory Method
- Builder
- Singleton
- Prototype





# Estructurales

- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy



# Comportamiento

- Command
- Chain of responsibility
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor



# Agenda

- Definiciones
- Clasificación
- Ejercicios



# Ejercicio 1

Se tiene un sistema con acceso a una base de datos. Dicha base sólo admite 10 conexiones simultáneas desde el sistema en cuestión, y se prefiere evitar cerrar conexiones que se van a volver a usar en un futuro cercano ya que establecer la conexión es un proceso costoso y lento.

## Ejercicio 2

Se ha desarrollado una herramienta tipo botonera, donde cada botón de la misma se podría asociar a diferentes acciones según la utilidad que se le quiera dar. Es necesario definir qué se va a hacer ante la acción de cada botón que se cree.

# Ejercicio 3

En el controlador de un portón motorizado, se definió que el portón tiene que responder a una única señal enviada desde un control remoto de un solo botón, elegido principalmente por cuestiones de costos. Según el caso, el controlador tendrá que decidir qué acción cambio debe efectuar sobre el portón.

- Si está cerrado, debe abrirlo
- Si está abierto, debe cerrarlo
- Si está cerrándose, debe volver a abrirlo
- Si está abriéndose, debe volver a cerrarlo

## Ejercicio 4

En la implementación de un nuevo sistema, se acordó la reutilización de una pieza de software *legacy* que cumple en la empresa cliente, una función determinada. La prestación es bastante compleja y rehacerla es inviable porque se desconocen los pormenores de su funcionamiento interno, y excede los costos anticipados para el proyecto en curso. Sin embargo, la interfaz que ofrece es más bien simple y está correctamente documentada.

## Ejercicio 5

Un microservicio que está implementando debe devolver para una consulta, 43 campos que componen la información básica de la entidad que representa. Dicho microservicio cuenta con una framework base que transforma cualquier objeto java en una representación serializada a JSON de dicho objeto antes de devolverlo, pero para aprovecharlo, debe convertir toda la información recabada de la base de datos en un único objeto que los contenga todos.



# Bibliografía

- Design Patterns CD - Gamma, Helm, Johnson, Vlissides

